Traitement Automatique des Langues Naturelles (TALN) [1]

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles.

Volume 1 : conférence principale

Pascal Denis, Natalia Grabar, Amel Fraisse, Rémi Cardon, Bernard Jacquemin, Eric Kergosien, Antonio Balvet (Éds.)

Lille, France, 28 juin au 2 juillet 2021

1. https://talnrecital2021.inria.fr/

Avec le soutien de

Soutiens institutionnels

Sponsors industriels

Partenaires « Argent »

Partenaires « Bronze »

Préface

Pour sa 28e édition, la conférence TALN s'est tenue pour la première fois de son histoire à Lille, et pour la seconde fois seulement dans la région des Hauts-de-France (après TALN 2009 à Senlis). Comme il en est devenu la tradition, TALN est une nouvelle fois organisée sous l'égide de l'ATALA conjointement avec sa conférence "soeur", RÉCITAL, dont c'est déjà la 23e édition.

Comme pour leurs éditions 2020, TALN 2021 et RÉCITAL 2021 ont à nouveau dû être "virtualisées" en raison de l'épidémie de Covid-19 qui a paralysé la France, l'Europe, et une bonne partie du monde. Ceci a considérablement compliqué son organisation et a conduit à la suppression de plusieurs événements originellement prévus dont le HackaTAL, la soirée gala, les événements sociaux, les promenades à Lille, la dégustation de la cuisine régionale, etc. Néanmoins, nous avons pu maintenir l'atelier Défi Fouilles de Textes (DEFT 2021), ainsi que non moins de 8 tutoriels différents. Nous remercions les organisateurs vaillants de DEFT et des tutoriels.

En lien avec cette actualité sanitaire, le thème choisi pour l'édition de TALN 2021 est "TAL et santé". Ce thème se reflète naturellement dans le programme de cette édition, puisqu'elle comprend une conférence invitée de Pierre Zweigenbaum sur le TAL médical, une session dédiée, et le traitement des cas cliniques comme tâche de DEFT 2021. Nous avons par ailleurs été très contents d'accueillir André Martins (professeur associé à l'Instituto Superior Técnico et VP recherche chez Unbabel, à Lisbonne au Portugal), comme second conférencier invité de cette édition.

Ces actes regroupent les articles des conférences TALN et RÉCITAL (volume 1 et 2, respectivement), ceux décrivant les démonstrations (volume 3), ceux issus de l'atelier DEFT 2021 (volume 4). Comme lors de la précédente édition de TALN 2020, un appel spécifique réservé aux résumés d'articles publiés dans des conférences internationales de premier plan fut également organisé. Ces résumés ont été versés dans le volume 1.

Pour TALN, un total de 58 articles a été soumis, soit exactement le même nombre que pour l'édition précédente. Parmi ceux-ci, 45 ont été sélectionnés, soit un taux d'acceptation de 77.6 %, dont 8 comme articles longs et 37 comme articles courts. Pour RÉCITAL, le nombre d'articles soumis fut de 16, en léger recul par rapport aux 22 soumissions de l'an dernier. 13 de ceux-ci ont été sélectionnés, soit un taux d'acceptation de 81.2 %.

Parmi les innovations de cette édition de TALN-RÉCITAL, nous avons rajouté une phase de discussion entre auteur(e)s et relecteures/relectrices, de manière à enrichir et fluidifier le processus de relecture et, on l'espère, à améliorer la séléction des articles et la plus-value des retours apportée aux auteur(e)s.

Nous sommes extrêmement reconnaissants à toutes les personnes qui ont participé aux différents comités scientifiques de ces conférences, à savoir :
— les responsables de domaine de TALN (voir page vi) ;
— les relectrices et relecteurs de TALN et RÉCITAL (voir page vi).
En outre, nous remercions chaleureusement l'ATALA, dont le comité permanent (le CPerm) assure la pérennité des TALN et RÉCITAL. Nous sommes également redevables à l'ensemble des membres du comité d'organisation (en particulier Antonio Balvet et Bernard Jacquemin), ainsi qu'aux personnes qui ont apporté leur soutien administratif et logistique

(en particulier Christine Yvoz) pour leur implication. Merci aussi à Yannick Parmentier qui nous a permis de produire ces actes et d'assurer la diffusion de ceux-ci sur HAL, l'ACL anthology et les archives TALN. Nous remercions aussi Onkar Pandit, Mariana Vargas et Nathalie Vauquier pour leur aide dans la maintenance du site web de la conférence et dans la configuration de la plate-forme `gather.town`.

Enfin, que soient aussi remerciés nos partenaires institutionnels et industriels pour leur soutien financier, en particulier : le CNRS, l'Inria, l'Université de Lille, les laboratoires CRIStAL, STL et GERIICO, l'ATALA et l'Afia, la GDLFLF, et les entreprises Schlumberger, ELRA, ERDIL, SINEQUA, ZENDOC.

Les présidentes et présidents de TALN : Pascal Denis et Natalia Grabar ;

Les présidentes et présidents de RÉCITAL : Amel Fraisse et Rémi Cardon.

Comités

Co-Président.e.s TALN

— Pascal Denis, MAGNET, Inria Lille & CRIStAL

— Natalia Grabar, STL, CNRS

Responsables de domaine

— Delphine Bernhard, LiLPA, Strasbourg

— Houda Bouamor, CMU Qatar

— Chloé Braud, IRIT, Toulouse

— Caroline Brun, NaverLabs, Grenoble

— Marie Candito, LLF, Paris

— Caio Corro, LISN, CNRS, Université Paris-Saclay

— Géraldine Damnati, Orange Labs, Lannion

— Maud Erhmann, EPFL, Suisse

— Cécile Fabre, CLLE, Toulouse

— Benoît Favre, TALEP, Marseille

— Thomas François, CENTAL, UCLouvain, Louvain-la-Neuve, Belgique

— Nuria Gala, LPL, Aix

— Philippe Langlais, DIRO, Montréal, Canada

— Philippe Muller, IRIT, Toulouse

— Alexis Nasr, TALEP, Marseille

— Magalie Ochs, LIS, Marseille

— Yannick Parmentier, LORIA, Nancy

— Tim van de Cruys, KUL, Leuven, Belgique

— Guillaume Wisniewski, LLF, Paris

Comité de lecture TALN

— Céline Alec, GREYC, Université de Caen-Normandie

— Alexandre Allauzen, LAMSADE, Université Paris-Dauphine

— Maxime Amblard, LORIA, Université de Lorraine

— Pascal Amsili, LATTICE, ILPGA, Université Sorbonne Nouvelle

— Loïc Barrault, University of Sheffield

— Patrice Bellot, Aix-Marseille Université – CNRS (LIS)

— Asma Ben Abacha, NLM/NIH, USA

— Laurent Besacier, Naver Labs Europe

— Yves Bestgen, F.R.S-FNRS et UCL

— Philippe Blache, LPL, CNRS

— Nathalie Camelin, LIUM, Le Mans Université

— Rémi Cardon, STL CNRS, Université de Lille

— Peggy Cellier, IRISA, INSA Rennes

— Thierry Charnois, LIPN, CNRS Université Sorbonne Paris Nord

— Vincent Claveau, IRISA, CNRS

— Maximin Coavoux, Université Grenoble Alpes, CNRS

— Mathieu Constant, ATILF, Université de Lorraine

— Benoit Crabbé, Université de Paris, LLF

— Béatrice Daille, LS2N, CNRS, Université de Nantes

— Mathieu Dehouck, CNRS, LATTICE

— Gaël Dias, Université de Normandie

— Patrick Drouin, OLST, Université de Montréal

— Emmanuelle Esperança-Rodier, LIG, Université Grenoble Alpes

— Dominique Estival, Western Sydney University

— Olivier Ferret, CEA List, Université Paris-Saclay

— Cyril Grouin, LISN, CNRS, Université Paris-Saclay

— Gaël Guibon, Télécom Paris et SNCF

— Olivier Hamon, Syllabs

— Thierry Hamon, Université Paris-Saclay, CNRS, LISN & Université Sorbonne Paris Nord

— Nabil Hathout, CLLE, CNRS

— Amir Hazem, LS2N, CNRS, Université de Nantes

— Nicolas Hernandez, LS2N, CNRS, Université de Nantes

— Stéphane Huet, LIA, Université d'Avignon

— Christine Jacquin, LS2N, CNRS, Université de Nantes

— Sylvain Kahane, Modyco, Université Paris Nanterre

— Mikaela Keller, MAGNET, Université Lille & CRIStAL

— Olivier Kraïf, LIDILEM, Université Grenoble Alpes

— Matthieu Labeau, Telecom Paris

— Éric Laporte, LIGM, Université Gustave Eiffel

— Gwénolé Lecorvé, Univ Rennes, CNRS, IRISA

— Benjamin Lecouteux, LIG, Université Grenoble Alpes

— Claire Lemaire, Lairdil, Université Paul Sabatier, Toulouse III ; LIG, Université Grenoble Alpes

— Yves Lepage, Université Waseda, Japon

— Cedric Lopez, EMVISTA

— Denis Maurel, Université de Tours, Lifat

— Anne-Lyse Minard, LLL, CNRS, Université d'Orléans

— Richard Moot, LIRMM, CNRS & Université de Montpellier

— Véronique Moriceau, IRIT, Université de Toulouse

— Emmanuel Morin, LS2N, CNRS, Université de Nantes

— Luka Nerima, LATL-CUI, Université de Genève

— Aurélie Névéol, LISN, CNRS, Université Paris-Saclay

— Jian-Yun Nie, Université de Montréal

— Damien Nouvel, ERTIM, INALCO

— Sylvain Pogodalla, LORIA, INRIA

— Jean-Philippe Prost, Aix-Marseille Université et Université de Montpellier

— Solen Quiniou, LS2N, CNRS, Université de Nantes

— Christian Raymond, IRISA, INSA Rennes

— Christian Retoré, LIRMM Univ Montpellier CNRS

— Sophie Rosset, LISN, CNRS, Université Paris-Saclay

— Didier Schwab, LIG, Université Grenoble Alpes

— Pascale Sébillot, IRISA, INSA Rennes

— Gilles Sérasset, LIG, Université Grenoble Alpes

— Ludovic Tanguy, CLLE, Université de Toulouse

— Xavier Tannier, Sorbonne Université, INSERM, LIMICS

— Andon Tchechmedjiev, EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales

— Charles Teissedre, SYNAPSE

— Juan-Manuel Torres-Moreno, Laboratoire Informatique d'Avignon / UA

— Nicolas Turenne, United International College, Chine

— François Yvon, LISN, CNRS, Université Paris-Saclay

— Pierre Zweigenbaum, LISN, CNRS, Université Paris-Saclay

Co-Président.e.s RECITAL

— Amel Fraisse (GERIICO)

— Rémi Cardon (STL)

Comité de lecture RECITAL

— Jean-Yves Antoine (Université François Rabelais de Tours)

— Sonia Badene (Linagora IRIT)

— Rachel Bawden (INRIA)

— Johana Bodard (THIM/CHart EA4004 – Université Paris 8)

— Chloé Braud (IRIT – CNRS)

— Johanna Mayra Cordova (INALCO)

— Núria Gala (Aix-Marseille Université, LPL CNRS)

— Mahault Garnerin (Université Grenoble Alpes)

— Loïc Grobol (Lattice)

— William Havard (Université Grenoble Alpes)

— Laurine Huber (LORIA)

— Mikaela Keller (Université de Lille – INRIA)

— Yves Lepage (Waseda University)

— Anne-Laure Ligozat (LISN, CNRS, Université Paris-Saclay, ENSIIE)

— Damien Nouvel (INALCO)

— Patrick Paroubek (LISN, CNRS, Université Paris-Saclay)

— Thierry Poibeau (LaTTiCe-CNRS)

— Laurent Romary (INRIA & HUB-ISDL)

— Nicolas Turenne (INRA UPEM)

— Zheng Zhang (Schlumberger, AI Lab)

— Pierre Zweigenbaum (LISN, CNRS, Université Paris-Saclay)

Table des matières

Première partie

Articles longs

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Auto-encodeurs variationnels : contrecarrer le problème de *posterior collapse* grâce à la régularisation du décodeur

Alban Petit Caio Corro
Universite Paris-Saclay, CNRS, LISN, 91400, Orsay, France
{alban.petit,caio.corro}@limsi.fr

RÉSUMÉ

Les auto-encodeurs variationnels sont des modèles génératifs utiles pour apprendre des représentations latentes. En pratique, lorsqu'ils sont supervisés pour des tâches de génération de textes, ils ont tendance à ignorer les variables latentes lors du décodage. Nous proposons une nouvelle méthode de régularisation fondée sur le *dropout* « fraternel » pour encourager l'utilisation de ces variables latentes. Nous évaluons notre approche sur plusieurs jeux de données et observons des améliorations dans toutes les configurations testées.

ABSTRACT

Variational auto-encoders : prevent posterior collapse via decoder regularization

Variational autoencoders are powerful generative models useful to learn latent representations. However, when supervised for text generation tasks, they tend to ignore the latent variables. We propose a novel regularization method based on fraternal dropout to encourage the use of latent variables. We evaluate our approach and observe improvements in all the tested configurations.

MOTS-CLÉS : auto-encodeurs variationnels, régularisation, génération automatique de textes.

KEYWORDS: variational auto-encoders, regularization, automatic text generation.

1 Introduction

Les modèles génératifs reposant sur une pondération neuronale comme les auto-encodeurs variationnels (Kingma & Welling, 2014, AEV) et les réseaux antagonistes (Goodfellow *et al.*, 2014, RA), entre autres, connaissent une grande popularité dans tous les domaines de l'apprentissage automatique dont le traitement automatique des langues (TAL). Les AEV peuvent manipuler des variables observées discrètes ce qui les rend particulièrement intéressants pour la génération de textes, contrairement aux RA. Dans ces modèles, la génération d'une phrase suit le processus suivant :

$$\boldsymbol{z} \sim p(\boldsymbol{z}), \quad x_1 \sim p_\theta(x_1|\boldsymbol{z}), \quad x_2 \sim p_\theta(x_2|\boldsymbol{z}, x_1) \quad x_3 \sim p_\theta(x_3|\boldsymbol{z}, x_1, x_2), \quad \dots$$

où $\boldsymbol{z} \in \mathbb{R}^k$ est une représentation (ou plongement) de phrase latente et $x_1, x_2, \cdots \in X$ les mots composant la phrase tirés d'un vocabulaire X. La génération se termine lorsqu'un mot spécial indiquant la fin de la phrase est généré. Sans perte de généralité, nous faisons l'hypothèse que la loi *a priori* $p(\boldsymbol{z})$ est pré-définie et non apprise. L'indice θ dénote les paramètres de la distribution conditionnelle et dans notre cas, θ correspond aux paramètres d'un réseau de neurones. Il est important de noter que nous ne faisons aucune hypothèse d'indépendance dans la distribution conditionnelle $p(x_t|\boldsymbol{z}, \boldsymbol{x}_{<t})$.

Volume 1 : conférence principale, pages 2–10.

Lors de la phase d'apprentissage, l'objectif est de calculer la valeur des paramètres θ pour maximiser la vraisemblance des données :

$$\max_{\theta} \mathbb{E}_{\tilde{p}(\boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x})\right] = \max_{\theta} \mathbb{E}_{\tilde{p}(\boldsymbol{x})} \left[\int \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}\right] = \max_{\theta} \mathbb{E}_{\tilde{p}(\boldsymbol{x})} \left[\mathcal{L}(\boldsymbol{x}, \theta)\right] \quad (1)$$

où $\tilde{p}$ est la distribution empirique des données d'entrainement. L'objectif est incalculable dans le cas général, sans même parler d'optimiser celui-ci, à cause de la marginalisation sur la représentation latente $\boldsymbol{z}$. Les méthodes variationnelles proposent d'introduire une loi de proposition $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ et de construire un objectif de substitution comme suit :

$$\mathcal{E}(\boldsymbol{x}, \theta, \phi) = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]}_{\text{reconstruction}} - \underbrace{\mathrm{KL}[q_\phi(\cdot|\boldsymbol{x})|p(\cdot)]}_{\text{divergence avec l'}\textit{a priori}} \quad (2)$$

où une borne variationnelle (l'*evidence lower bound* ou ELBO), notée $\mathcal{E}$, peut être utilisée pour optimiser une borne sur l'objectif de l'équation 1 car $\forall \phi : \mathcal{E}(\boldsymbol{x}, \theta, \phi) \leq \mathcal{L}(\boldsymbol{x}, \theta)$. L'objectif de l'entraînement devient alors :

$$\max_{\theta, \phi} \mathbb{E}_{\boldsymbol{x} \sim \tilde{p}(\boldsymbol{x})} \left[\mathcal{E}(\boldsymbol{x}, \theta, \phi)\right] \quad (3)$$

où l'optimisation se fait à la fois sur les paramètres θ et sur les paramètres de la distribution de proposition ϕ. L'algorithme *Expectation-Maximization* (Dempster *et al.*, 1977, EM) résout ce problème en optimisant l'objectif par blocs, c'est à dire en optimisant successivement l'objectif en fonction de ϕ (étape E) et θ (étape M). Contrairement à EM, l'approche dite des AEV consiste à optimiser ce problème par montée de gradient stochastique jointe sur ϕ et θ. De plus, contrairement aux applications standards d'EM, ni la distribution ni la famille de la *posterior* $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ ne sont connues. Il est donc fait l'hypothèse d'indépendance sur les coordonnées de $\boldsymbol{z}$ dans la distribution q_ϕ. Enfin, la distribution q_ϕ est paramétrée par un réseau de neurones et apprise sur l'ensemble des données. Notons que lors de l'entrainement, le terme de reconstruction dans l'équation 2 est approximé via Monte-Carlo en utilisant un seul échantillon. Il est donc usuel d'appeler la distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ qui génère une représentation latente à partir d'une phrase comme l'**encodeur** et la distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ qui reconstruit la phrase original comme le **décodeur**.

Malheureusement, en pratique, les AEV pour la génération automatique de textes sont sensibles au phénomène d'effondrement de l'*a posteriori* (Bowman *et al.*, 2016, *posterior collapse*). Informellement, cela signifie que la loi de proposition n'est pas optimisée correctement et reste proche de la distribution *a priori* pour tous les points : $\forall \boldsymbol{x} : q_\phi(\boldsymbol{z}|\boldsymbol{x}) \simeq p(\boldsymbol{z})$. Cela conduit à une très mauvaise approximation de l'objectif 1 par la ELBO. *In fine*, le décodeur va ignorer cette variable latente et donc aucune représentation de phrase $\boldsymbol{z}$ n'est apprise.

De nombreux travaux se sont focalisés sur ce problème de *posterior collapse*. Nous pouvons les diviser en deux catégories. **(1)** D'une part, des modifications de la fonction objectif ont été proposées. (Bowman *et al.*, 2016) propose d'introduire une pondération du terme de divergence qui permet de tempérer son importance durant l'entrainement. (Kingma *et al.*, 2016) et (Pelsmaeker & Aziz, 2020) ont proposé d'ajouter des contraintes sur le terme de divergence qui obligent ce dernier à être supérieur à un hyperparamètre. Enfin, d'autres travaux ont proposé des fonctions de pertes alternatives pour entrainer des AEV dans le cadre de la génération de textes (Livne *et al.*, 2020; Havrylov & Titov, 2020), entre autres. **(2)** D'autre part, des travaux ont proposé de modifier l'architecture du décodeur. (Yang *et al.*, 2017) ont proposé de remplacer le réseau de neurones récurrents dans le décodeur par un réseau convolutif. (Dieng *et al.*, 2019) proposent d'utiliser des connexions *skip* entre la représentation latente et les différentes couches cachées du décodeur. Ces connexions forment des

« raccourcis » entre la variable latente et les couches de sorties afin de promouvoir cette première. L'idée est d'utiliser des architectures neuronales qui se reposent davantage sur la variable latente et donc amoindrissent indirectement l'impact de la divergence avec la distribution *a priori* sur l'objectif.

Dans ce travail, nous proposons une nouvelle approche pour contrecarrer le problème d'effondrement de l'*a posteriori*, fondée sur la régularisation des décodeurs, c'est à dire que nous ne modifions ni la fonction objectif ni la structure du décodeur. Nos contributions peuvent être résumées comme suit :

— Nous proposons d'utiliser la régularisation des paramètres pour contrecarrer l'impact de l'effondrement de l'*a posteriori*. Nous nous focalisons sur le *dropout* « fraternel » (Zolna *et al.*, 2018) pour obliger le décodeur à utiliser la représentation latente.

— Nous expérimentons notre approche dans différentes configurations en utilisant les travaux de (Li *et al.*, 2019) comme point de référence. Nous soulignons que nous ne changeons pas les hyperparamètres de leur modèle et les réutilisons tels que distribués dans le code source pour ne pas biaiser les résultats en faveur de notre approche.

Nous espérons que cet article va encourager de futurs travaux à explorer cette nouvelle piste d'amélioration pour la génération automatique de textes fondée sur les AEV.

2 Régularisation du décodeur : le *dropout* « fraternel »

Afin d'éviter le problème d'effondrement de la distribution *a posteriori*, une idée est d'augmenter artificiellement l'importance du gradient de l'encodeur provenant du décodeur. C'est cette intuition qui est utilisée dans l'architecture proposée par (Dieng *et al.*, 2019). Le LSTM (Hochreiter & Schmidhuber, 1997), un réseau neuronal récurrent, est une architecture efficace qui peut minimiser le terme de reconstruction tout en ignorant la valeur de la variable latente. C'est notamment l'objectif qui est atteint par les modèles de langues à l'état de l'art. Notre approche consiste à régulariser les paramètres du LSTM afin que les représentations cachées à chaque étape soient moins dépendantes des mots donnés en entrée. Dans ce cas, le décodeur est forcé d'utiliser la représentation cachée, donnant lieu à un gradient plus important à l'encodeur. Pour cela, nous proposons d'utiliser le *dropout* « fraternel » (Zolna *et al.*, 2018).

Le terme de reconstruction dans la fonction objectif des AEV est un terme de maximisation de la log-vraisemblance utilisant la technique habituelle de supervision forcée pour les modèles de langue : lors de l'apprentissage, ce modèle autorégressif est entrainé à prédire le mot suivant en fonction de la séquence de mots précédemment observée dans les données. Soit $\boldsymbol{x} \in X^n$ une phrase de longueur n. Chaque mot x_i est représenté par un vecteur provenant d'une table de plongements lexicaux. Nous désignons l'ensemble de ces vecteurs par une matrice $\boldsymbol{E} \in \mathbb{R}^{w \times n}$ où w est la dimension des plongements lexicaux. Une représentation contextuelle est calculée pour chaque position en utilisant le LSTM :

$$\boldsymbol{H} = \textsc{Lstm}(\boldsymbol{E}, \boldsymbol{z}; \theta)$$

où $\boldsymbol{H} \in \mathbb{R}^{d \times n}$ est une matrice contenant les états cachés de dimension d en sortie du LSTM paramétré par θ, pour chaque mot de la phrase. La variable latente $\boldsymbol{z}$ est projetée puis donnée à la fois comme initialisation de la mémoire et de l'état caché. Elle est également concaténée à chaque entrée. Nous reportons le lecteur à (Li *et al.*, 2019) pour plus de détails sur l'architecture et les différents paramètres.

Le *dropout* de plongements lexicaux (Dozat & Manning, 2017) consiste à remplacer aléatoirement certains plongements par un vecteur de 0 lors de l'entrainement pour éviter le sur-apprentissage. Le

calcul des représentations cachées devient alors :

$$\boldsymbol{d} \sim \mathcal{B}(b), \qquad \boldsymbol{E}' \in \mathbb{R}^{w \times n} \text{ t.q. } E'_{i,j} = E_{i,j} d_j, \qquad \boldsymbol{H} = \textsc{Lstm}(\boldsymbol{E}', \boldsymbol{z}; \theta)$$

où $\boldsymbol{d} \in \{0,1\}^n$ est un vecteur de booléens dont chaque élément est tiré indépendemment d'une Bernoulli de paramètre $b \in [0,1]$. La matrice $\boldsymbol{E}'$ correspond à la matrice $\boldsymbol{E}$ où les éléments de certaines colonnes ont été remplacés par des 0. Le *dropout* « fraternel » consiste quant à lui à créer deux matrices $\boldsymbol{E}'$ et $\boldsymbol{E}''$ de la façon suivante :

$$\begin{aligned}
&\boldsymbol{d} \sim \mathcal{B}(b), \\
&\boldsymbol{E}' \in \mathbb{R}^{d \times n} \text{ t.q. } E'_{i,j} = E_{i,j} d_j, && \boldsymbol{E}'' \in \mathbb{R}^{d \times n} \text{ t.q. } E''_{i,j} = E_{i,j}(1 - d_j), \\
&\boldsymbol{H}' = \textsc{Lstm}(\boldsymbol{E}', \boldsymbol{z}; \theta), && \boldsymbol{H}'' = \textsc{Lstm}(\boldsymbol{E}'', \boldsymbol{z}; \theta).
\end{aligned}$$

Les matrices $\boldsymbol{E}'$ et $\boldsymbol{E}''$ sont ensuite utilisées pour calculer deux fois la log-vraisemblance de la phrase, leur moyenne remplaçant le terme initial dans le terme de reconstruction. Un terme de régularisation est ensuite ajouté à la fonction objectif présentée dans l'équation 3 :

$$\mathcal{R}(\theta; \alpha) = -\alpha \| \textsc{Lstm}_\theta(\boldsymbol{E}') - \textsc{Lstm}_\theta(\boldsymbol{E}'') \|_2^2$$

où $\alpha > 0$ est un hyperparamètre. Pour que ce terme soit maximisé, le modèle devra utiliser l'information commune aux deux décodages, c'est à dire la représentation latente $\boldsymbol{z}$.

3 Expériences

Pour évaluer notre solution, nous avons utilisé le code source et les métriques implémentés par (Li *et al.*, 2019) auquel nous avons ajouté notre méthode de régularisation. Nous avons conservé les mêmes hyperparamètres que ceux distribués par les auteurs pour ne pas biaiser les résultats en notre faveur.

Nous nous évaluons sur deux jeux de données : Yelp (Shen *et al.*, 2017) et les données Stanford Natural Language Inference (Bowman *et al.*, 2015, SNLI). Ces deux jeux de données ont été sous-échantillonnés pour contenir 100 000 phrases d'entrainement et 10 000 pour la validation et le test chacun. SNLI et Yelp ont des vocabulaires respectivement de taille 9 990 et 8 411 et ont 10 mots par phrase en moyenne.

3.1 Métriques d'évaluation

Nous utilisons différentes métriques pour évaluer la qualité des modèles probabilistes appris.

(Log-vraisemblance et perplexité par mot) La log-vraisemblance négative (NLL) $-\mathbb{E}_{q_\phi(\boldsymbol{z}|s)} p_\theta(s|\boldsymbol{z})$ indique à quel point le modèle parvient à reconstruire l'entrée. La perplexité par mot (PPL) est la moyenne géométrique de l'inverse de la probabilité assignée au bon mot par le modèle. Puisqu'on observe l'opposé de la log-vraisemblance dans le premier cas et l'inverse des probabilités dans le second, on souhaite minimiser ces deux valeurs. Ces deux métriques sont approximés via 100 échantillons tirés de la distribution q pour chaque phrase.

(Score BLEU) Pour chaque phrase du jeu de test, une représentation latente est échantillonnée dans la distribution q puis une phrase est générée sans guider le modèle à partir de cette représentation. Le

α	NLL ↓	PPL ↓	UA ↑	IM ↑	BLEU ↑
0.01	**28.91**	**18.44**	4	6.34	5.20
0.1	29.50	19.12	5	7.03	6.40
0.5	30.69	21.53	**6**	**7.47**	**6.81**
1.0	32.30	25.28	4	6.87	6.03
2.0	33.41	28.27	4	7.29	6.09

TABLE 1 – Impact de l'hyperparamètre α du *dropout* fraternel sur les données Yelp. On veut maximiser les valeurs suivies du symbole ↑ et minimiser celles suivies par ↓.

score BLEU (Papineni *et al.*, 2002) va représenter la proportion de n-grammes (pour n allant de 1 à 4) de la phrase générée que l'on retrouve effectivement dans la phrase source. Si la phrase générée est plus courte que la phrase source, une pénalité est appliquée car il est plus simple de ne pas faire d'erreurs quand moins de mots sont générés.

(Unités actives) Le nombre d'unités actives (UA) indique le nombre de dimensions de la variable latente qui co-varient avec les observations. D'après (Burda *et al.*, 2016), un plus grand nombre d'unités actives est généralement représentatif d'une représentation latente plus riche. Comme dans leur article, une unité est considérée active si $Cov(s, \mathbb{E}_{q_\phi(\boldsymbol{z}|s)}[\boldsymbol{z}]) > 0.01$. Dans la suite de cet article, la dimension des variables latentes étant 32, nous aurons au maximum 32 unités actives.

(Information mutuelle) Nous reportons également l'information mutuelle entre la variable latente et la distribution de sortie. Une information mutuelle plus élevée indiquera que la variable latente est mieux utilisée par le modèle. Nous suivons la méthodologie de (He *et al.*, 2019).

3.2 Baseline

Nous évaluons notre approche en la comparant à plusieurs références, incluant un AEV « standard ». Toutes nos expériences utilisent une pondération du terme de divergence lors de l'optimisation du modèle. Cette technique proposée par (Bowman *et al.*, 2016) consiste à faire varier progressivement le facteur de pondération de 0 jusqu'à 1 lors des premières itérations complètes de l'entraînement. Cela impose au modèle de ne pas tenir compte de ce terme pendant les premières étapes de l'apprentissage, qui se focalisent donc sur l'apprentissage de la représentation.

Bits gratuits (Kingma *et al.*, 2016) La technique des bits gratuits consiste à ajouter une contrainte sur le terme de divergence avec l'*a priori* pour que celui-ci ne tombe pas en dessous d'une valeur pré-définie λ. Nous fixons $\lambda = 8$ dans toutes nos expériences.

Pré-entrainement (Li *et al.*, 2019) Cette solution propose d'entrainer d'abord le modèle comme un auto-encodeur standard. Ensuite, le décodeur est réinitialisé puis le modèle est entrainé comme un AEV.

3.3 Résultats et analyses

Nous reportons l'impact de l'hyperparamètre α du *dropout* « fraternel » sur les données Yelp dans la table 1. Nous observons qu'il existe un compromis entre les différentes métriques d'évaluation. Il s'agit donc ici de trouver un point d'équilibre entre la minimisation de la NLL et de la PPL et la maximisation des UA, de l'IM et du score BLEU. Dans la suite des expériences, nous avons fixé $\alpha = 0.1$ qui semble être une valeur raisonnable.

	Yelp					SNLI				
Configuration	NLL ↓	PPL ↓	UA ↑	IM ↑	BLEU ↑	NLL ↓	PPL ↓	UA ↑	IM ↑	BLEU ↑
Standard	33.40	28.25	2	1.14	1.43	32.57	20.64	**3**	0.52	2.32
+ dropout fraternel	**29.50**	**19.12**	**5**	**7.03**	**6.40**	**30.01**	**16.28**	2	**4.75**	**5.73**
Bits gratuits	29.54	19.20	32	5.69	4.02	28.88	14.66	32	4.63	4.77
+ dropout fraternel	**25.46**	**12.76**	32	**8.65**	**11.23**	**27.92**	**13.40**	32	**7.11**	**8.44**
Pré-ent.	33.74	29.21	2	0.71	0.83	31.76	19.14	3	1.14	2.76
+ dropout fraternel	**26.18**	**13.71**	**22**	**8.24**	**9.69**	**24.69**	**9.92**	**22**	**8.32**	**13.43**
Bits gratuits + Pré-ent.	25.93	13.37	32	8.14	7.54	23.33	8.75	32	8.49	13.9
+ dropout fraternel	**23.63**	**10.62**	32	**8.81**	**13.54**	**21.00**	**7.04**	32	**9.07**	**21.35**

TABLE 2 – Résultats des différentes métriques sur Yelp et SNLI pour quatres variantes de VAE avec et sans *dropout* fraternel. On veut maximiser les valeurs suivies du symbole ↑ et minimiser celles suivies par ↓.

Bits gratuits + pré-ent.
a boy is in front of a group of people.
a man in a blue shirt is standing in front of a crowd of people.
a child in blue is holding a camera.
a child in blue pants holding a camera while another man watches.
a child in blue pants holding a camera while another man in a black shirt looks on.
Bits gratuits + pré-ent. + dropout fraternel
the young boy is in a picture.
the young child is in front of a mother.
the small child is in front of a mother.
a small child in pink holds a picture of her mother.
a small child in pink sits in a picture with her mother.

TABLE 3 – Exemples d'interpolations entre deux représentations échantillonnées *a priori* pour les configurations avec un pré-entrainement et les bits gratuits sur SNLI. La deuxième configuration inclue aussi le dropout fraternel.

Nous reportons les résultats de quatre configurations de (Li *et al.*, 2019) avec et sans le *dropout* fraternel » dans le tableau 2. Notre approche apporte des améliorations pour toutes les métriques pour tous les modèles sur les deux jeux de données. Le phénomène d'effondrement de l'*a posteriori* est important sur les configurations n'utilisant ni bits gratuits ni *dropout* « fraternel », l'information mutuelle étant autour de 1 et le nombre d'unités actives étant de 2. L'ajout du *dropout* « fraternel » permet de gagner entre 4 et 7 points d'information mutuelle dans les deux configurations. Dans la configuration où le modèle est pré-entrainé, on observe que le pré-entrainement seul n'empêche pas l'effondrement de la distribution *a posteriori* puisqu'on voit 2 et 3 UA respectivement alors que le modèle conserve 22 unités actives avec le *dropout* « fraternel ». De façon intéressante, notre approche a un impact supérieur en comparaison aux bits gratuits pour toutes les métriques sauf le score UA sur Yelp ainsi que sur l'IM et le score BLEU sur SNLI : ceci peut indiquer que cette dernière approche force artificiellement les variables latentes à être décorrélées les unes des autres sans pour autant avoir l'impact escompté sur les autres métriques, car le *dropout* « fraternel » atteint les mêmes mesures avec seulement 5 et 2 unités actives respectivement sur Yelp et SNLI.

Comme expliqué précédemment, le score BLEU est calculé sur des phrases générées sans guider le modèle et permet donc aussi d'estimer la qualité des représentations latentes. Encore une fois, ce score est significativement meilleur lorsque l'on introduit le *dropout* « fraternel ».

Interpolation Nous reportons sur la Table 3 des exemples de phrases générées via interpolation entre deux échantillons de l'*a priori*. Nous observons que notre méthode semble générer des phrases cohérentes avec une évolution progressive de la longueur et du sens entre les différentes phrases.

4 Conclusion

Dans cet article, nous proposons de régulariser le décodeur pour contrecarrer le problème de l'effondrement de l'*a posteriori* lors de l'entrainement d'un AEV. Cette approche est différente de ce qui a été exploré dans la littérature. Nous observons qu'elle atteint ses deux objectifs qui sont d'améliorer la qualité de la génération de texte et surtout d'accroître l'utilisation de la variable latente. Des travaux futurs pourront s'intéresser à l'utilisation d'autres méthodes de régularisation des réseaux de neurones récurrents (Kanuparthi *et al.*, 2019; Krueger *et al.*, 2016; Gal & Ghahramani, 2016).

Remerciements

Nous remercions les 3 relecteurices anonymes pour leurs remarques et suggestions. Nous remercions François Yvon et Matthieu Labeau pour les relectures. Ces travaux ont bénéficié de calculs réalisés sur la plateforme Saclay-IA et d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 20XX-AD011011600 attribuée par GENCI.

Références

BOWMAN S. R., ANGELI G., POTTS C. & MANNING C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)* : Association for Computational Linguistics.

BOWMAN S. R., VILNIS L., VINYALS O., DAI A., JOZEFOWICZ R. & BENGIO S. (2016). Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, p. 10–21, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/K16-1002.

BURDA Y., GROSSE R. B. & SALAKHUTDINOV R. (2016). Importance weighted autoencoders. In Y. BENGIO & Y. LECUN, Éds., *Proceedings of 4th International Conference on Learning Representations*.

DEMPSTER A. P., LAIRD N. M. & RUBIN D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, **39**(1), 1–22.

DIENG A. B., KIM Y., RUSH A. M. & BLEI D. M. (2019). Avoiding latent variable collapse with generative skip models. In K. CHAUDHURI & M. SUGIYAMA, Éds., *Proceedings of Machine Learning Research*, volume 89 de *Proceedings of Machine Learning Research*, p. 2397–2405 : PMLR.

DOZAT T. & MANNING C. D. (2017). Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* : OpenReview.net.

GAL Y. & GHAHRAMANI Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In D. LEE, M. SUGIYAMA, U. LUXBURG, I. GUYON & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 29, p. 1019–1027 : Curran Associates, Inc.

GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A. C. & BENGIO Y. (2014). Generative adversarial nets. In Z. GHAHRAMANI, M. WELLING, C. CORTES, N. D. LAWRENCE & K. Q. WEINBERGER, Éds., *Advances in Neural Information Processing Systems 27 : Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, p. 2672–2680.

HAVRYLOV S. & TITOV I. (2020). Preventing posterior collapse with levenshtein variational autoencoder. *arXiv preprint arXiv :2004.14758*.

HE J., SPOKOYNY D., NEUBIG G. & BERG-KIRKPATRICK T. (2019). Lagging inference networks and posterior collapse in variational autoencoders. In *Proceedings of the 7th International Conference on Learning Representations* : OpenReview.net.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

KANUPARTHI B., ARPIT D., KERG G., KE N. R., MITLIAGKAS I. & BENGIO Y. (2019). h-detach : Modifying the LSTM gradient towards better optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* : OpenReview.net.

KINGMA D. P., SALIMANS T., JOZEFOWICZ R., CHEN X., SUTSKEVER I. & WELLING M. (2016). Improved variational inference with inverse autoregressive flow. In D. D. LEE, M. SUGIYAMA, U. V. LUXBURG, I. GUYON & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 29*, p. 4743–4751. Curran Associates, Inc.

KINGMA D. P. & WELLING M. (2014). Auto-encoding variational bayes. In Y. BENGIO & Y. LECUN, Éds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

KRUEGER D., MAHARAJ T., KRAMÁR J., PEZESHKI M., BALLAS N., KE N. R., GOYAL A., BENGIO Y., LAROCHELLE H., COURVILLE A. C. & PAL C. (2016). Zoneout : Regularizing rnns by randomly preserving hidden activations. *CoRR*, **abs/1606.01305**.

LI B., HE J., NEUBIG G., BERG-KIRKPATRICK T. & YANG Y. (2019). A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3603–3614, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1370.

LIVNE M., SWERSKY K. & FLEET D. J. (2020). Sentencemim : A latent variable language model. *arXiv preprint arXiv :2003.02645*.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318 : Association for Computational Linguistics. DOI : 10.3115/1073083.1073135.

PELSMAEKER T. & AZIZ W. (2020). Effective estimation of deep generative language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7220–7236, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.646.

SHEN T., LEI T., BARZILAY R. & JAAKKOLA T. (2017). Style transfer from non-parallel text by cross-alignment. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 30, p. 6830–6841 : Curran Associates, Inc.

YANG Z., HU Z., SALAKHUTDINOV R. & BERG-KIRKPATRICK T. (2017). Improved variational autoencoders for text modeling using dilated convolutions. In D. PRECUP & Y. W. TEH, Éds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 de *Proceedings of Machine Learning Research*, p. 3881–3890, International Convention Centre, Sydney, Australia : PMLR.

ZOLNA K., ARPIT D., SUHUBDY D. & BENGIO Y. (2018). Fraternal dropout. In *International Conference on Learning Representations*.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Biais de genre dans un système de traduction automatique neuronale : une étude préliminaire

Guillaume Wisniewski[1] Lichao Zhu[1] Nicolas Ballier[2] François Yvon[3]
(1) LLF, Univ. Paris & CNRS , 75013, Paris France
(2) CLILLAC-ARP, Univ. Paris, 750013, Paris France
(3) LISN, Univ. Paris-Saclay & CNRS , 91403, Orsay France
{guillaume.wisniewski,lichao.zhu,nicolas.ballier}@u-paris.fr,
francois.yvon@limsi.fr

RÉSUMÉ

Cet article présente les premiers résultats d'une étude en cours sur les biais de genre dans les corpus d'entraînements et dans les systèmes de traduction neuronale. Nous étudions en particulier un corpus minimal et contrôlé pour mesurer l'intensité de ces biais dans les deux directions anglais-français et français-anglais ; ce cadre contrôlé nous permet également d'analyser les représentations internes manipulées par le système pour réaliser ses prédictions lexicales, ainsi que de formuler des hypothèses sur la manière dont ce biais se distribue dans les représentations du système.

ABSTRACT

Gender Bias in Neural Translation : a preliminary study

This paper is a blueprint of a current study in the making on gender bias in French/English neural translation toolkits. We discuss previous research using probes for neural machine translation. We then study a minimal controlled corpus and use it to measure the intensity of such biases in the two translation directions (from and into English). Using a controlled experimental design also enables us to analyze the internal representations (attention matrices) of the translation system, and to formulate hypotheses regarding the way these biases are encoded within these representations.

MOTS-CLÉS : biais de genre, traduction automatique neuronale, évaluation diagnostique en TAL.

KEYWORDS: Gender bias, Neural Machine Translation, Diagnostic Evaluation in NLP.

1 Introduction

Il est largement admis (Callison-Burch *et al.*, 2006; Lo & Wu, 2011; Balvet, 2020) que les métriques automatiques classiques telles que les scores BLEU (Papineni *et al.*, 2002) ou METEOR (Banerjee & Lavie, 2005) sont inadaptées pour rendre compte des progrès observables en matière de qualité des traductions automatiques (TA) prédites par les systèmes neuronaux. Partant de ce constat, plusieurs protocoles (Isabelle *et al.*, 2017; Burlot & Yvon, 2017, 2018) ont été récemment proposés pour évaluer et diagnostiquer plus finement la traduction entre l'anglais et le français. Ces protocoles reposent sur l'utilisation de jeux de tests élaborés (manuellement ou automatiquement) pour confronter les systèmes de TA à des problèmes de traduction spécifiques et bien caractérisés.

Les limitations de la traduction automatique ne se réduisent pas à leur incapacité à prendre en charge

certains phénomènes linguistiques ; un autre problème important est l'existence de biais systématiques, en particulier de genre. Sous cette appellation, il faut distinguer plusieurs traits problématiques : (a) le fait que des erreurs de traductions sont plus fréquentes pour des énoncés qui mettent en scène des participantes de genre féminin ; (b) le fait que des traductions rendent linguistiquement explicites le genre des actants évoqués, alors que l'intention du locuteur peut être de le laisser ambigu ; (c) le fait que ces explicitations privilégient des assignations stéréotypiques, confortant, voire renforçant des préjugés sexistes dans les textes traduits. Dans la typologie de Crawford, affinée par (Blodgett *et al.*, 2020), ces problèmes sont susceptibles de fausser la manière dont certains groupes (ici, les femmes) sont représentés dans les textes *(representational harm)* ainsi que de conduire à un service (de TA) de moindre qualité pour les femmes *(allocational harm)*. Avec la massification de l'usage des technologies de TA, l'existence de tels biais est de plus en plus criante et dénoncée ; répondre à ces dénonciations exige à la fois des études précises (voir en particulier (Savoldi *et al.*, 2021) et les références citées), et des réponses idoines de la part des fournisseurs de technologie [1]. Ces questions sont en particulier discutées dans les actes de la série d'ateliers sur les biais de genre en traitement des langues [2].

Pour la paire de langues anglais-français, ces problèmes peuvent être mis en évidence et quantifiés en observant la manière dont les marques de genre, qui peuvent être explicites ou non dans le texte source, se distribuent dans le texte cible. Une mesure de cet effet, mis en évidence dans plusieurs travaux, est proposée par (Stanovsky *et al.*, 2019), qui évalue les biais de genre à partir du décompte des erreurs portant sur la résolution d'anaphores pronominales.

La première contribution de cet article est d'étendre les analyses conduites sur la traduction depuis l'anglais vers le français dans la direction inverse, en proposant de nouveaux contrastes pour mettre en évidence et quantifier ces biais de genre. Nous nous intéressons, dans un second temps, à partir de l'analyse des représentations internes d'un système de traduction neuronale à base de TRANSFORMER, à identifier plus finement la manière dont ces biais sont encodés dans les paramètres du réseau, en particulier ceux qui servent aux calculs des matrices d'attention.

2 Un jeu de tests contrôlé pour observer les biais de genre

Dans cette section, nous présentons la démarche qui nous a conduit à construire de nouveaux contrastes pour observer et quantifier les biais de genre en TA.

2.1 Les corpus WinoGender et WinoBias

Notre point de départ est l'étude de Stanovsky *et al.* (2019), qui formule des propositions concrètes pour évaluer les biais de genre, en s'appuyant principalement sur deux jeux de données : Winogender [3] (Rudinger *et al.*, 2018) et WinoBias [4] (Zhao *et al.*, 2018), tous deux inspirés des schémas Winograd

1. À titre d'illustration, les efforts de Google pour remédier à ces effets sont décrits dans ce billet https://ai.googleblog.com/2020/04/a-scalable-approach-to-reducing-gender.html.
2. Gender bias in NLP, http://genderbiasnlp.talp.cat.
3. https://github.com/rudinger/winogender-schemas
4. https://www.aclweb.org/anthology/attachments/N18-2003.Datasets.zip

(Winograd, 1983) [5]. Un schéma Winograd repose sur une paire de phrases, chacune composée de deux propositions, qui ne diffèrent que d'un seul mot (ou une expression) prédicatif. Changer le verbe dans le prédicat induit un changement dans l'interprétation de la coréférence dans la subordonnée, qui renvoie au sujet ou à l'objet de la principale comme dans l'exemple suivant :

(1) *The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.*

Dans cet exemple, la première partie de l'alternative conduit à interpréter *they* comme référant à *The city councilmen*, alors que la seconde induit une coréférence avec *the demonstrators*. Ces schémas constituent des cas de test particulièrement difficiles pour les systèmes de TAL, car la résolution correcte de l'anaphore implique souvent une analyse profonde, voire des connaissances du monde.

Rudinger *et al.* (2018) décalquent ce schéma d'alternance pour 120 couples de phrases en mobilisant deux types de constructions pour constituer le corpus **Winogender** :
— *[entity1] [interacts with] [entity2] [conjunction] [pronoun] [circumstances].*
— *[entity1] [interacts with] [entity2] and then [interacts with] [pronoun] for [circumstances].*
Les jeux de tests qui en dérivent reposent alors sur l'établissement de la relation de coréférence entre *he/she* et son antécédent dans des phrases comme (le référent attendu est entre crochets) :

(2) *[The developer] built a website for the tailor because [she] is an expert in building websites.*

(3) *The developer built a website for [the tailor] because [he] wants to sell cloths online.*

WinoBias est construit sur des principes similaires et comprend un ensemble équilibré de 3 160 phrases contenant des anaphores pronominales dont l'antécédent est un nom d'activité ou de profession. L'association pronom/nom est également répartie entre (a) des situations « stéréotypiques » (conformes aux distributions par genre de ces activités dans la population) et non-stéréotypiques ; (b) des structures dans lesquelles l'anaphore peut être résolue à partir de la syntaxe, et des structures pour lesquelles il faut des connaissances supplémentaires.

2.2 Une évaluation des biais de genre

(Stanovsky *et al.*, 2019) utilise les 3 880 phrases issues de ces travaux pour mesurer les biais de systèmes traduisant depuis l'anglais vers 8 langues dans lesquelles le genre est grammaticalisé.

L'exemple (2) ci-dessus correspond à une situation non-stéréotypique et sera jugé correct si *developer* est traduit par *développeuse*, incorrect sinon. Selon ces auteurs, le biais se manifeste par des erreurs de traduction qui privilégient des genres associés à des rôles stéréotypiques plutôt que ceux qui sont attendus au vu de la relation de coréférence. Ils proposent donc de les mesurer en comparant les taux d'erreurs des traductions des pronoms associés respectivement à des noms masculins et féminins.

La méthode de (Stanovsky *et al.*, 2019) pour mesurer le biais est problématique à plusieurs titres, et finalement peu appropriée pour notre étude. Outre son utilisation d'un corpus artificiel, on note qu'elle repose sur un repérage automatique du genre du nom choisi par le système ; or ce repérage n'est pas toujours possible (quand le groupe nominal ne porte aucune marque de genre), comme lorsque l'on traduit la phrase suivante dans lequel le genre du groupe nominal sujet ne peut être déterminé :

5. On se reportera à (Levesque *et al.*, 2012) pour une discussion de ces schémas et à Amsili & Seminck (2017) pour leur adaptation au français.

(4) *L'analyste se dirigea vers la caissière car elle devait faire un dépôt.*

Ce repérage peut également être bruité du fait d'erreurs dans l'étiquetage morphosyntaxique ou dans l'alignement automatique avec la phrase anglaise, ou encore du fait d'une traduction très erronée [6].

Sémantiquement, l'absence d'alternance de prédicat peut conduire à des ambiguïtés de la portée du pronom. À la différence des énoncés des tests de Winograd, dont la robustesse de l'interprétation fait l'objet de tests d'évaluation par des humains, les énoncés de WinoGender n'ont pas été testés et sont parfois ambigus entre ce qu'on appelle parfois une interprétation *de dicto* ou *de re* : ainsi, dans l'exemple ci-dessous, la proposition introduite par *because* peut expliciter soit le point de vue de l'énonciateur *(de dicto)* soit celui de l'actant (*de re*) :

(5) *The chief gave the housekeeper a tip because she was satisfied*

(6) *The chief gave the housekeeper a tip because he was helpful.*

Dans cet exemple, l'alternance prévue de la coréférence dans ce modèle de test n'est ainsi pas certaine : dans la phrase (6), *he* pourrait renvoyer à l'objet *(de re)* ou au sujet grammatical *(de dicto)*, de sorte que l'alternance en genre de *chief* n'est pas garantie dans ce couple de phrases.

Un second problème est que ce test est difficile à « inverser » pour évaluer ces phénomènes dans la direction français-anglais. Nos premières tentatives pour construire un jeu de test en post-éditant des traductions automatiques de WinoGender se sont rapidement heurtées à de nombreux cas d'ambiguïté dans la détermination du genre correct français. Il apparaît enfin que ce corpus contient un trop grand nombre de sources de variabilité (des structures de phrase et du lexique) pour que l'on puisse facilement exploiter les matrices d'attention calculées pendant la traduction. Nous avons préféré utiliser un jeu de données plus simple dans nos expériences, en nous inspirant des travaux de Saunders & Byrne (2020) qui sont présentés à la section 5.2.

2.3 Une évaluation plus contrôlée du biais de genre

À l'instar de Saunders & Byrne (2020), nous avons construit un ensemble équilibré de 388 phrases en anglais sur le patron *The [noun] completed [his/her] work*, où [noun] est un nom de profession. Dans ces phrases, la seule marque de genre est alors portée par le pronom [7] *her/his*.

Chaque patron est instancié une fois au masculin et une fois au féminin. La mesure des performances repose sur le calcul du genre du GN traduisant [the noun] en français et qui se trouve toujours en début de phrase. Quatre cas sont possibles, selon que le genre est porté par l'article et le nom (*la traductrice*), seulement le nom (*l'actrice*), seulement l'article (*la juge*), ou complètement ambigu (*l'analyste*). Contrairement aux données de Stanovsky *et al.* (2019), évaluer la correction des traductions est ici facile, car la position des mots portant l'information de genre est toujours la même.

Nous avons également traduit automatiquement ces phrases en français puis corrigé / normalisé les traductions pour construire le test (vers l'anglais) sur le modèle *[det] [nom] a fini son travail*. Les noms de profession en français ont été vérifiés à partir des listes de référence (Becquer *et al.*, 1999;

6. Ainsi, les trois résultats du Tableau 1 qui portent sur les 3 880 exemples de WinoGender, excluent chacun plusieurs centaines de phrases (près de 900 pour le système `fairseq`), pour lesquelles le script d'analyse échoue à prédire le genre.

7. Nous suivons ici (Huddleston *et al.*, 2002) qui voient dans l'anglais une langue où le genre est peu grammaticalisé mais présent dans les relations de coréférence, comme avec les réfléchis *himself / herself / itself*

Dister & Moreau, 2014). La vérification en anglais de la traduction correcte s'appuie sur le simple repérage du pronom (*her*/*his*) dans la phrase cible. Dans cette direction, le genre sera soit déduit de celui du groupe nominal sujet en français (inférence du genre nom ou du déterminant), soit reflétera une préférence dans l'association [nom] / [genre du pronom] en anglais.

Une version alternative de ces tests remplace [the] (en anglais) par *each* et [det] en français par l'épicène *chaque*, ceci afin qu'en français la seule marque de genre soit (éventuellement) sur le nom, ce qui simplifie l'analyse des matrices d'attention qui ont alors une forme encore plus régulière. L'ensemble des corpus ainsi construits est librement téléchargeable à partir de l'URL : https://github.com/neuroviz/neuroviz/tree/main/gender_analysis_in_mt.

3 Expérimentations et résultats globaux

3.1 Le système de traduction

Nous avons utilisé l'outil JoeyNMT, qui propose une implémentation « pédagogique » d'un système de traduction à base de Transformer (Vaswani *et al.*, 2017) permettant d'obtenir des résultats proches de l'état de l'art (Kreutzer *et al.*, 2019). Dans notre système, encodeur et décodeur sont composés de 6 couches, chacune avec 8 têtes d'attention ; les couches de *feed-forward* comportent 2 048 paramètres et la dimension des plongements lexicaux est 512. Notre modèle comportait, au total, 76 596 736 paramètres. Le système a été entraîné avec les données de la tâche « News » de la campagne WMT'15 [8]. Les corpus `Europarl`, `NewsCommentary` et `CommonCrawl` sont utilisés pour l'apprentissage, regroupant 4 813 682 phrases et près de 141 millions de mots français. Tous les corpus ont été convertis en minuscules, tokenisés et segmentés en unités sous-lexicales en utilisant le modèle unigramme de l'outil *SentencePiece* (Kudo, 2018) ; le vocabulaire résultant contient 32 000 unités. Le modèle est entraîné en optimisant l'entropie croisée à l'aide de la stratégie Adam. Ce système obtient sur le corpus `newstest2014` un score BLEU de 34,0 (resp. 32,7) pour la direction français-anglais (resp. anglais-français).

Un autre point de comparaison est donné dans le Tableau 1, qui reproduit pour ce système les mesures de biais de genre de (Stanovsky *et al.*, 2019), en les comparant avec deux systèmes considérés dans cette étude, celui de `fairseq` (Ott *et al.*, 2018) et des traductions réalisées avec le système de Systran. [9] Il apparaît que notre implémentation de JoeyNMT délivre des performances conformes à celles des autres systèmes pour la prédiction du genre, avec une forte différence avec les prédictions pour le masculin et le féminin, et donc un fort biais de genre.

3.2 Évaluation de la traduction du genre

Nous évaluons la capacité d'un système à prédire le genre des métiers à partir du corpus décrit § 2. Pour la traduction vers l'anglais, cette évaluation est simple et repose sur la vérification du genre du pronom *her*/*his*. Vers le français, le genre du groupe nominal peut être marqué soit par le déterminant,

8. Il s'agit de la dernière campagne d'évaluation sur la paire anglais-français organisée dans le cadre de la conférence WMT (http://statmt.org/wmt15).

9. Dans ces deux derniers cas, nous utilisons les traductions de Stanovsky *et al.* (2019) et renvoyons à cette référence pour une description plus précise de ces deux systèmes.

JoeyNMT		Fairseq		Systran	
Acc	ΔG_s	Acc	ΔG_s	Acc	ΔG_s
45,6	30,1	48,0	4,4	43,4	41,8

TABLE 1 – Évaluation de notre système de traduction sur les phrases de WinoGender. Pour chaque système, nous calculons l'exactitude (*accuracy*) de la prédiction du pronom, ainsi que la différence de scores F1 entre la prédiction des phrases pour les genres masculin et féminin.

soit par le nom. Sauf mention contraire, nous considérerons que le genre du GN est correctement prédit lorsque le genre du déterminant *et* le genre du nom sont tous deux corrects.

Il faut noter qu'il n'est pas toujours possible de déterminer l'information de genre dans les traductions prédites par un système de TA. En effet, dans certains cas, le système produit une traduction correcte n'utilisant pas les pronoms *her/his* (p. ex. *the programmer has finished working*) ; dans d'autres cas la traduction est complètement fausse (p. ex. « l'inspectrice a fini son travail. » a été traduit en « *the young man bent on to work.* ») ou le déterminant est traduit par *its* (53 phrases correspondant pour la plupart à des situations où le nom de métier n'a pas été traduit correctement).

3.3 Résultats expérimentaux

Appliqué au corpus décrit à la section 2, dans le sens français-anglais, notre système prédit correctement le genre du pronom possessif anglais dans 65,7% des cas. Dans le sens anglais-français, il traduit correctement le genre du GN dans 46,1% des cas (respectivement 60,5% des cas pour le genre du nom, et 55,6% pour le genre du déterminant). Ces résultats [10] médiocres montrent qu'un système neuronal «standard» a des difficultés à modéliser et à prédire les informations de genre au cours de la traduction. Pour les mettre en perspective, nous avons également utilisé, pour traduire ces mêmes corpus, un moteur « grand public » `DeepL` (version 1.12.0) et `e-translation` [11], moteur de traduction développé par la Commission Européenne et librement accessible à des fins de recherche académique. Si les résultats de DeepL sont meilleurs (voir le tableau 2), ils restent très imparfaits et montrent que la tâche proposée est difficile. À notre grande surprise, les traductions de `e-translation` ne font que très peu d'erreur dans la traduction du genre, ce qui laisse supposer que ce système intègre un traitement spécifique de ces phénomènes.

Les résultats détaillés sont dans le tableau 2. Ils montrent que les systèmes considérés présentent des taux d'erreurs très différents entre genres, avec des écarts marqués entre systèmes et directions de traduction. Cette observation justifie d'étudier simultanément les deux directions. Pour notre système de traduction, la plupart des erreurs de prédiction portent sur le féminin : par exemple, pour la traduction vers l'anglais, le taux d'erreur pour les pronoms féminins est de 70,67% contre 2,87% pour les pronoms masculins.

À l'inverse, DeepL privilégie quasi systématiquement le féminin lorsqu'il traduit vers l'anglais : 252 des prédictions du système contiennent *her*, quand seulement 141 contiennent *his* (dans de rares cas, le système propose également l'alternative *his or her*). Ce comportement est inversé pour la

10. Ces calculs ignorent les hypothèses de traduction pour lesquelles aucun genre ne peut être déterminé. Pour le système français-anglais, cela arrive pour 16% des sorties, qui ne contiennent ni *her* ni *his*. Pour le système anglais-français, le genre du GN sujet n'a pu être déterminé dans 17% des cas

11. ec.europa.eu/cefdigital/eTranslation

dét. (fr)	nom (fr)	fr → en			en → fr		
		JoeyNMT	DeepL	e-translation	JoeyNMT	DeepL	e-translation
l'	épicène	46,3	53,2	68,8	77,3	85,4	84,4
	féminin	16,0	93,3	100	4,3	10,0	21,4
	masculin	90,9	54,0	100	67,7	94,6	86,1
la	épicène	26,4	94,9	100	0,0	15,4	58,3
	féminin	62,8	96,5	98,1	2,0	22,0	37,3
le	épicène	95,1	65,9	100	87,0	95,7	88,6
	masculin	100,0	70,1	100	82,0	98,9	92,5

TABLE 2 – Pourcentage de succès dans le transfert du genre entre français et anglais. La 2ème colonne distingue les cas où le nom de métier est genré en français et la première indique le déterminant du nom de métier (ces valeurs sont déterminées sur la référence pour la traduction de l'anglais vers le français).

traduction vers le français : dans ce cas, les noms de métiers sont presque toujours traduits par un masculin.

4 Analyses de la propagation de l'information de genre

Dans cette section, nous présentons plusieurs analyses complémentaires portant sur la manière dont l'information de genre est propagée depuis le GN en français vers le pronom anglais. Notre principale objectif est de déterminer quelles sont les éléments mis en jeu dans le choix du pronom *his*/*her*. En particulier, nous nous intéressons à la manière dont les représentations et les différents scores d'attention sont influencés par les informations de genre.

Rappelons que dans une architecture transformer standard, trois mécanismes attentionnels sont simultanément à l'œuvre : l'auto-attention de l'encodeur, qui permet que les représentations des tokens sources s'influencent mutuellement ; l'auto-attention du décodeur, qui joue un rôle similaire côté cible, sous la contrainte que chaque mot n'a accès qu'aux représentations des mots qui le précèdent ; enfin l'attention croisée source-cible dans le décodeur, qui permet de contextualiser les représentations cibles en les combinant avec les représentations source sur la dernière couche de l'encodeur. Nous nous intéressons ici principalement à l'auto-attention de l'encodeur.

4.1 Impact du déterminant du GN

Pour mesurer l'impact du déterminant du GN sujet, nous avons réalisé une première expérience de contrôle en construisant un nouveau corpus de test identique à celui construit à la section 2 mais dans lequel nous neutralisons tous les déterminants dont la forme varie avec le genre : dans ce corpus, les déterminants des noms de métiers ont été systématiquement remplacés par le déterminant épicène *chaque*. Le genre à transférer n'est alors marqué que sur le nom de métier et, donc indéterminé quand ce dernier est épicène. Dans cette configuration, notre système ne commet aucune erreur en transférant en anglais le genre du nom de métier masculin, alors qu'il se trompe presque systématiquement (94,55 % d'erreurs) pour les féminins.

4.2 Le genre de *son*

La question que nous étudions dans cette section porte sur le transfert de l'information de genre entre langues. Pour traduire correctement le genre du GN français, trois hypothèses (non mutuellement exclusives) sont envisagées : (a) une influence *directe* par le calcul de l'attention cross-lingue effectuée lors de la traduction du pronom ; (b) une influence *indirecte* passant par l'encodage (cross-lingue) du genre dans la représentation du nom anglais, qui est propagée vers le pronom ; (c) une influence *indirecte* passant par l'encodage (monolingue) du genre dans la représentation du possessif français *son*, qui est ensuite propagée (cross-lingue) vers le pronom anglais. Ces trois possibilités sont résumées dans la figure 1. Nous nous intéressons à valider ou invalider l'existence du mécanisme (c), en analysant de plusieurs manières la représentation de *son*. Ce choix est motivé par la structure systématique des phrases françaises, qui ont recours au même mot, qui de surcroît se trouve à la même position, et dont la représentation est donc facile à extraire et manipuler.

Sonder *son* La première méthode repose sur l'utilisation de sondes linguistiques (*probes*) (Belinkov & Glass, 2019) et consiste à tester la capacité de prédire le genre du GN en observant seulement la représentation du mot *son* construite par le système de TA. En utilisant l'encodeur du système de traduction, nous calculons le vecteur de représentation associé à ce mot pour les 388 phrases de notre corpus. Nous entraînons ensuite un classifieur linéaire simple qui doit prédire le genre du GN à partir du seul vecteur représentant le possessif : l'hypothèse est que s'il est possible de réaliser avec succès cette prédiction, c'est que les représentations de *son* pour les phrases comportant un GN masculin diffèrent de celles qui comportent un GN féminin, et pourront donc influencer utilement le choix du pronom en anglais.

Expérimentalement, nous apprenons un modèle de régression logistique avec `scikit-learn` (Pedregosa *et al.*, 2011) en utilisant 75% des données, et calculons les taux d'erreurs sur les 25% restant. Cette expérience est répétée 100 fois pour pouvoir calculer l'intervalle de confiance de la précision. Compte tenu du rapport entre la taille des vecteurs d'entrée (512) et le nombre de données d'apprentissage (388), il est nécessaire de régulariser fortement ce classifieur, ce que nous obtenons en ajoutant à la fonction objectif une pénalité ℓ_1 [12]. D'une manière générale, il importe de contrôler la capacité des sondes, et de s'assurer qu'elles ne pourraient pas également apprendre des étiquetages aléatoires (Hewitt & Liang, 2019).

Les résultats sont rapportés dans le tableau 3. On constate que l'information de genre est effectivement présente, mais uniquement dans les couches les plus profondes : la précision du classifieur utilisant comme caractéristique les représentations de `son` issues des deux premières couches est très faible (proche de celle d'un classifieur prenant ses décisions au hasard) mais augmente rapidement (de plus de 20 points entre la 2ème et la 4ème couche) pour se stabiliser autour de 80%. Suivant les recommandations de (Hewitt & Liang, 2019) nous avons également calculé la précision de notre sonde après avoir appliqué une permutation aléatoire des étiquettes afin de nous assurer que la sonde ne capturait pas des corrélations fallacieuses. Les résultats rapportés à la table 3 montre que les informations de genre sont bien présentes dans la représentation et non dans la sonde.

Manipuler *son* La seconde méthode que nous proposons utilise *une intervention* pour mettre en évidence ce même effet. Elle consiste à remplacer le vecteur représentant ce mot à la sortie de

12. Au final, dans nos expériences entre 30% et 80% des paramètres sont nuls.

Couche	Précision sonde	Précision aléatoire
1	$57,4\%$ ± 0.8	$45,3\%$ ± 0.9
2	$60,4\%$ ± 1.0	$50,7\%$ ± 0.8
3	$72,5\%$ $\pm 0,8$	$48,8\%$ ± 0.9
4	$82,0\%$ $\pm 0,6$	$48,6\%$ ± 0.8
5	$81,9\%$ $\pm 0,7$	$49,6\%$ ± 0.8
6	$79,3\%$ $\pm 0,7$	$49,2\%$ ± 0.8

TABLE 3 – Précision d'un classifieur prédisant le genre du GN à partir de la représentation de `son`.

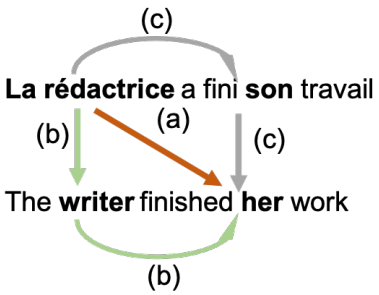


FIGURE 1 – Les différents éléments pouvant influencer le choix du genre du pronom possessif en anglais.

l'encodeur alternativement par (a) une version *neutre*, obtenue en moyennant les représentations de *son* sur l'ensemble des phrases (notre corpus est construit avec la contrainte que le nombre de GN féminin est le même que le nombre de GN masculin) ; (b) une représentation *masculine* (resp. *féminine*) supposée prototypique, obtenues en encodant respectivement les deux phrases suivantes :

(7) le facteur a terminé son travail.

(8) la pharmacienne a terminé son travail.

Ces deux phrases ont été choisies parce que, dans les deux cas, la traduction du genre par notre système est correcte et parce que l'information du genre est portée à la fois par le déterminant et par le nom. Le reste de la traduction se déroule sans autre modification, ce qui nous permet de comparer l'influence de ces 4 représentations (trois fixes, une contextuelle) sur la traduction du pronom.

Les résultats figurent dans le tableau 4 : nous y avons rapporté la proportion d'hypothèses de traduction dans lesquelles le pronom possessif est féminin, masculin ou ne peut pas être déterminé, en fonction de l'intervention sur la représentation de `son`. Contrairement à ce qui était attendu, modifier la représentation du pronom possessif n'a que peu d'impact sur le choix du pronom possessif *his* ou `her`. Ces résultats montrent que la représentation de `son` construite par le système de TA n'est pas (ou peu) utilisée lors de la génération de l'hypothèse de traduction, bien que les résultats rapportés dans le paragraphe précédent montrent que celles-ci sont particulièrement pertinentes. Ce résultat contre intuitif rejoint plusieurs observations dans la littérature des sondes : ce n'est pas parce qu'une information « linguistique » est encodée dans les représentations neuronales qu'elle est exploitée par le réseau (Belinkov & Glass, 2019).

4.3 Analyse des matrices d'attention

Un des intérêts du corpus proposé dans ce travail est de faciliter l'analyse des matrices d'attention au cœur des systèmes de traduction neuronaux. En effet, les phrases ayant une structure fixe, il est aisé d'identifier la position des mots marquant le genre (p. ex. pour la traduction depuis le français du déterminant et du nom de métier) et d'analyser l'attention depuis ou vers ces positions. Nous présentons quelques résultats préliminaires illustrant le type d'études réalisables à partir de ce corpus.

intervention	genre de la traduction	% erreur	intervention	genre de la traduction	% erreur
aucune	féminin	10,6%	moyenne	féminin	10,4%
	indéterminé	13,4%		indéterminé	13,0%
	masculin	76,0%		masculin	76,6%
féminin	féminin	12,5%	masculin	féminin	11,2%
	indéterminé	10,6%		indéterminé	14,0%
	masculin	76,9%		masculin	74,8%

TABLE 4 – Manipulation des représentations de *son* : proportion d'hypothèses de traduction dans lesquelles le pronom possessif est féminin, masculin ou ne peut pas être déterminé, en fonction de l'intervention sur la représentation de `son`.

	couche n° 0		couche n° 1		couche n° 2		couche n° 3		couche n° 4		couche n° 5	
↓ préd. / vers →	dét.	mét.	dét.	mét.	dét.	mét.	dét.	mét.	dét.	mét.	dét.	mét.
correcte	0.018	0.111	0.069	0.134	0.178	0.114	0.174	0.072	0.147	0.151	0.228	0.203
incorrecte	0.017	0.074	0.063	0.137	0.173	0.114	0.181	0.081	0.153	0.140	0.226	0.211

TABLE 5 – Score d'auto-attention moyen entre le possessif *son* et le déterminant du nom de métier (dét.) ou le nom de métier (mét.), suivant que le genre soit prédit correctement ou non.

Pour mieux comprendre les informations utilisées pour prédire le genre du pronom anglais, nous représentons dans le tableau 5, la moyenne sur le corpus du score d'auto-association entre « *son* » et les deux mots du GN. Le modèle ayant 8 têtes d'attention par couches, nous avons considéré, pour chaque phrase, uniquement le plus grand score, ce qui revient à vérifier qu'au moins une tête pointe sur un des mots permettant de prédire le genre. Lorsque le nom de métier est segmenté en plusieurs unités sous-lexicales, seul le plus grand score d'attention vers une de ces unités est conservé."

Les résultats du tableau 5 montrent que les deux positions permettant de traduire correctement le mot « *son* » sont utilisées pour construire la représentation de ce mot : dans les deux dernières couches, le score d'attention entre «*son*» et ces deux mots dépasse 0,2 [13]. On note que ces scores sont comparables dans les traductions « correctes » et « erronées », à l'exception de l'attention vers le nom de métier sur la première couche. Ceci confirme les résultats présentés supra : le modèle s'appuie plus sur le nom de métier pour prendre ses décisions que sur l'article.

5 Travaux connexes : mesurer et corriger les biais de genre

5.1 Compter les erreurs et mesurer les biais

La première étape pour étudier les biais de genre en TA consiste à les caractériser plus précisément, ainsi que les effets néfastes qu'ils peuvent produire auprès des utilisateurs de cette technologie (Blodgett *et al.*, 2020). Ces auteurs distinguent en particulier les *biais de représentation*, qui conduiraient une TA à générer des textes véhiculant une représentation dénaturée des catégories sociales évoquées dans les textes traduits ; des *biais d'allocation*, qui se manifestent par un fonctionnement dégradé

13. Les scores d'auto-attentions sont positifs et normalisés de façon que toutes les auto-attentions entre un mot et les autres mots de la phrase somment à 1 ; dans la mesure où les phrases du corpus sont constitués en moyenne de 7 mots et que l'auto-attention entre un mot et lui-même est toujours élevée, une valeur de 0,2 peut être considérée comme importante.

(des systèmes) pour certaines catégories d'usagers.

Lorsque l'on aborde ces questions sous l'angle quantitatif, à partir des observables que sont les sorties des systèmes de TA, deux situations sont à distinguer. Dans la première, le genre des personnes dont il est fait mention dans un texte source à traduire est indéterminé [14] et ne peut être déduit du contexte ; dans ce cas, on doit souhaiter que la traduction conserve cette ambiguïté, car tout autre choix impliquerait une interprétation non conforme aux intentions de l'auteur, tout en constatant que l'expression de cette ambiguïté est plus ou moins directe et transparente selon les langues, qui pour certaines disposent de formes neutres, ou bien ne marquent qu'exceptionnellement le genre, quand d'autres le marquent obligatoirement. À défaut, il semble souhaitable que les marques de genre qui seraient insérées le soient de manière équilibrée [15]. Lorsque ce n'est pas le cas, le système risque de créer, voire d'amplifier les biais de représentation, de fournir des informations faussées aux utilisateurs de la TA et de les propager dans les étapes de traitement ultérieures.

La seconde situation est celle dans laquelle l'information de genre [16] est explicite dans le texte source, auquel cas il est attendu qu'elle soit transférée correctement dans le texte cible, afin toujours de préserver les intérêts de l'auteur ainsi que celui des personnes qui seraient évoquées dans le texte. De nouveau, le système peut commettre deux types d'erreurs : (i) introduire dans le texte cible une ambiguïté qui est absente de la source ; (ii) se tromper dans l'expression du genre correct (complètement ou partiellement — ce qui est possible quand le même genre est marqué sur plusieurs éléments du discours). En particulier entre dans cette catégorie le fait ne pas préserver l'ambiguïté ou la fluidité du genre alors que des pronoms sont disponibles pour éviter des assignations de genre binaire (voir pour l'anglais l'article de synthèse de (Cao & Daumé III, 2019)).

Même s'il est possible d'imaginer des situations dans lesquelles une traduction *fidèle* pourrait porter préjudice à certains usagers, il semble utile de mesurer les biais d'un système par des décomptes d'erreurs qu'il commet et la méthode que nous avons présentée supra s'inscrit dans cette démarche.

Pour effectuer ces décomptes, la plupart des travaux analysant les biais de genre dans la traduction neuronale se sont concentrés sur le lexique de la profession (Kuczmarski & Johnson, 2018; Prates *et al.*, 2019), en étudiant aussi bien des corpus artificiels que des corpus réels (Gonen & Webster, 2020). Notons que la question du genre en TA peut être abordée sous d'autres angles : ainsi, Vanmassenhove *et al.* (2018) présente des observations portant sur la distribution des verbes d'opinion en fonction du genre et du degré d'assertivité présumé chez les hommes et les femmes. Comme le montrent ces auteurs, qui étudient la traduction de 10 langues vers le français, enrichir la phrase source (en anglais) par l'information explicite du genre du locuteur permet alors d'obtenir des traductions meilleures qu'un système qui ne dispose pas de cette information.

Une tentative de mesurer les biais dans la traduction *vers l'anglais* est détaillée par Cho *et al.* (2019), qui élaborent un indice du biais dans la traduction depuis le coréen *(translation gender bias index)*. Cet indice évalue la propension d'un système à traduire un pronom neutre en coréen en un masculin ou un féminin en anglais, ou bien encore en un groupe nominal non marqué pour le genre.

14. Cette formulation est simplificatrice, puisque, par exemple, il a longtemps été accepté en français dans certains usages que le genre masculin ait une valeur de générique — dans cette situation, il faudrait considérer que le genre des personnes représentées est indéterminé, alors même qu'une marque explicite de genre est présente.

15. Il est toutefois possible d'imaginer des situations ou des applications qui justifieraient de favoriser un genre (linguistique) plutôt qu'un autre dans les sorties.

16. Qu'elle soit encodée sous la forme d'une catégorisation binaire du genre ou bien qu'elle corresponde à des assignations plus fluides des identités de genre.

5.2 Atténuer automatiquement les biais de genre

Mesurer les biais permet aussi d'évaluer l'impact de travaux visant à les atténuer dans des traductions automatiques. Ces travaux mobilisent principalement trois types de techniques (voir (Savoldi *et al.*, 2021) pour une étude récente). Une première consiste à manipuler les représentations lexicales. Elle est utilisée par Escudé Font & Costa-jussà (2019) qui injectent dans le système OpenNMT des plongements lexicaux entraînés avec l'algorithme *gender-neutral GloVe* de Zhao *et al.* (2018). Ils testent ensuite la capacité à désambiguïser *friend* dans les traductions vers l'espagnol à partir des relations de coréférence ainsi que d'un nom de profession en attribut dans des phrases de la forme *I've known her for a long time, my friend works as a refrigeration mechanic.*

Les techniques de pré-annotation (Sennrich *et al.*, 2016) insèrent dans le texte source des marques explicites de genre, qui vont servir à orienter le système vers des traductions correctes. C'est, par exemple, l'approche suivie par Vanmassenhove *et al.* (2018), qui montrent que l'indiciation du genre des entités nommées dans l'anglais ("*FEMALE Madam President, as a...*") permet d'améliorer les scores BLEU pour des traductions vers le français, l'italien, le danois et le finnois. Des résultats similaires sont obtenus par Basta *et al.* (2020) pour la direction anglais-espagnol et des analyses complémentaires sont réalisées par Saunders *et al.* (2020). Cette technique est enfin utilisée par Kuczmarski & Johnson (2018) pour contrôler la traduction *vers l'anglais* de formes pronominales non-marquées en turc dans des phrases telles que *"O bir doktor"* ou *"O bir hemşire"*.

Une troisième famille d'approches manipule les distributions des données d'apprentissage en s'appuyant sur des méthodes d'augmentation de données *(counterfactual data augmentation (CDA)*). Ainsi, Lu *et al.* (2020) engendrent automatiquement des corpus artificiels qui rétablissent l'équilibre en genre. Poursuivant cette direction, Saunders & Byrne (2020) montrent qu'il est plus simple et plus efficace de manipuler les distributions d'apprentissage en s'appuyant sur des méthodes d'adaptation au domaine. Ils utilisent un petit corpus artificiel équilibré en genres qui sert à adapter un système entraîné sur un corpus déséquilibré. Leur analyse de la traduction depuis l'anglais de trois langues montre que l'adaptation réduit les biais mesurés par les méthodes de Stanovsky *et al.* (2019).

6 Perspectives et Conclusions

Nous avons introduit dans ce travail un nouveau jeu de test permettant de mettre en évidence les biais de genre dans les systèmes de traduction automatique. Ce jeu de tests offre de nombreuses possibilités pour analyser finement les échanges d'informations entre les différentes composantes du réseau de neurones que nous souhaitons explorer dans nos travaux futurs. Nous pensons en effet, qu'en plus de leur quantification, une meilleure compréhension des causes des biais sont une étape nécessaire à la « neutralisation » de ceux-ci.

Remerciements

Ce travail a été partiellement financé par le projet NeuroViz / Explorations et visualisations d'un système de traduction neuronale, soutenu par la Région Ile-de-France dans le cadre d'un financement DIM RFSI 2020.

Références

AMSILI P. & SEMINCK O. (2017). Schémas Winograd en français : une étude statistique et comportementale. In *TALN 2017*, p. 28–35, Orléans, France. HAL : hal-01628342.

BALVET A. (2020). Métriques d'évaluation en traduction automatique : le sens et le style se laissent-ils mettre en équation ? In T. MILLIARESSI, Éd., *La Traduction épistémique : entre poésie et prose*, p. 315–356. Presses Universitaires du Septentrion.

BANERJEE S. & LAVIE A. (2005). METEOR : An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation*, p. 65–72, Ann Arbor, Michigan.

BASTA C., COSTA-JUSSÀ M. R. & FONOLLOSA J. A. R. (2020). Towards mitigating gender bias in a decoder-based neural machine translation model by adding contextual information. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*, p. 99–102, Seattle, USA : Association for Computational Linguistics. DOI : 10.18653/v1/2020.winlp-1.25.

BECQUER A., CERQUIGLINI B., CHOLEWKA N., COUTIER M., FRÉCHER J. & MATHIEU M.-J. (1999). *Femme, j'écris ton nom...* La Documentation française.

BELINKOV Y. & GLASS J. (2019). Analysis Methods in Neural Language Processing : A Survey. *Transactions of the Association for Computational Linguistics*, **7**, 49–72. DOI : 10.1162/tacl_a_00254.

BLODGETT S. L., BAROCAS S., DAUMÉ III H. & WALLACH H. (2020). Language (technology) is power : A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 5454–5476, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.485.

BURLOT F. & YVON F. (2017). Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, p. 43–55, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/W17-4705.

BURLOT F. & YVON F. (2018). Évaluation morphologique pour la traduction automatique : adaptation au français. In *Actes de la Conférence TALN. Volume 1-Articles longs, articles courts de TALN*, p. 61–74.

CALLISON-BURCH C., OSBORNE M. & KOEHN P. (2006). Re-evaluating the role of BLEU in machine translation research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

CAO Y. T. & DAUMÉ III H. (2019). Toward gender-inclusive coreference resolution. arXiv preprint http://arxiv.org/abs/1910.13913.

CHO W. I., KIM J. W., KIM S. M. & KIM N. S. (2019). On measuring gender bias in translation of gender-neutral pronouns. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, p. 173–181, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-3824.

DISTER A. & MOREAU M.-L. (2014). *Mettre au féminin : guide de féminisation des noms de métier, fonction, grade ou titre*. Fédération Wallonie-Bruxelles, 3e édition édition.

ESCUDÉ FONT J. & COSTA-JUSSÀ M. R. (2019). Equalizing gender bias in neural machine translation with word embeddings techniques. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, p. 147–154, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-3821.

GONEN H. & WEBSTER K. (2020). Automatically identifying gender issues in machine translation using perturbations. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 1991–1995, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.findings-emnlp.180.

HEWITT J. & LIANG P. (2019). Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 2733–2743, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1275.

HUDDLESTON R., PULLUM G. K. *et al.* (2002). *The Cambridge Grammar of English*. Cambridge University Press.

ISABELLE P., CHERRY C. & FOSTER G. (2017). A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 2486–2496, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/D17-1263.

KREUTZER J., BASTINGS J. & RIEZLER S. (2019). Joey NMT : A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) : System Demonstrations*, p. 109–114, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-3019.

KUCZMARSKI J. & JOHNSON M. (2018). Gender-aware natural language translation. *Technical Disclosure Commons*, p. 1–9.

KUDO T. (2018). Subword regularization : Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 66–75, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1007.

LEVESQUE H., DAVIS E. & MORGENSTERN L. (2012). The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

LO C.-K. & WU D. (2011). MEANT : An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 220–229.

LU K., MARDZIEL P., WU F., AMANCHARLA P. & DATTA A. (2020). Gender bias in neural natural language processing. In *Logic, Language, and Security*, p. 189–202. Springer. DOI : 10.1007/978-3-030-62077-6_14.

OTT M., EDUNOV S., GRANGIER D. & AULI M. (2018). Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation : Research Papers*, p. 1–9, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-6301.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, p. 311–318.

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.

PRATES M. O., AVELAR P. H. & LAMB L. C. (2019). Assessing gender bias in machine translation : a case study with Google translate. *Neural Computing and Applications*, p. 1–19.

RUDINGER R., NARADOWSKY J., LEONARD B. & DURME B. V. (2018). Gender bias in coreference resolution. In M. A. WALKER, H. JI & A. STENT, Éds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, p. 8–14 : Association for Computational Linguistics. DOI : 10.18653/v1/n18-2002.

SAUNDERS D. & BYRNE B. (2020). Reducing gender bias in neural machine translation as a domain adaptation problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7724–7736, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.690.

SAUNDERS D., SALLIS R. & BYRNE B. (2020). Neural machine translation doesn't translate gender coreference right unless you make it. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, p. 35–43, Barcelona, Spain (Online) : Association for Computational Linguistics.

SAVOLDI B., GAIDO M., BENTIVOGLI L., NEGRI M. & TURCHI M. (2021). Gender bias in machine translation. arxiv preprint http://arxiv.org/abs/2104.06001.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 35–40, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1005.

STANOVSKY G., SMITH N. A. & ZETTLEMOYER L. (2019). Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 1679–1684, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1164.

VANMASSENHOVE E., HARDMEIER C. & WAY A. (2018). Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 3003–3008, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1334.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 30*, p. 5998–6008 : Curran Associates, Inc.

WINOGRAD T. (1983). *Language as a cognitive process : Volume 1 : Syntax*. Addison-Wesley Pub. Co., Reading, MA.

ZHAO J., WANG T., YATSKAR M., ORDONEZ V. & CHANG K.-W. (2018). Gender bias in coreference resolution : Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 15–20, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-2003.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Exploration des relations sémantiques sous-jacentes aux plongements contextuels de mots

Olivier Ferret
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
olivier.ferret@cea.fr

RÉSUMÉ

De nombreuses études ont récemment été réalisées pour étudier les propriétés des modèles de langue contextuels mais, de manière surprenante, seules quelques-unes d'entre elles se concentrent sur les propriétés de ces modèles en termes de similarité sémantique. Dans cet article, nous proposons d'abord, en nous appuyant sur le principe distributionnel de substituabilité, une méthode permettant d'utiliser ces modèles pour ordonner un ensemble de mots cibles en fonction de leur similarité avec un mot source. Nous appliquons d'abord cette méthode pour l'anglais comme mécanisme de sondage pour explorer les propriétés sémantiques des modèles ELMo et BERT du point de vue des relations paradigmatiques de WordNet et dans le contexte contrôlé du corpus SemCor. Dans un second temps, nous la transposons à l'étude des différences entre ces modèles contextuels et un modèle de plongement statique.

ABSTRACT

Exploring semantic relations underlying contextual word embeddings.

Many studies were recently done for investigating the properties of contextual language models but surprisingly, only a few of them focus on the properties of these models in terms of semantic similarity. In this article, we first propose a method that exploits, by relying on the distributional principle of substitutability, these models for ranking a set of target words according to their similarity with a source word. Then, we apply this method for English as a probing mechanism for investigating the semantic properties of ELMo and BERT models from the viewpoint of WordNet's paradigmatic relations and in the controlled context of SemCor. Finally, we adapt and apply this method to the study of differences between these contextual models and static embeddings.

MOTS-CLÉS : Modèles de langue contextuels, sémantique distributionnelle, relations sémantiques.

KEYWORDS: Contextual language models, distributional semantics, semantic relations.

1 Introduction

L'introduction de plongements contextuels de mots tels qu'ELMo (Peters *et al.*, 2018) ou BERT (Devlin *et al.*, 2019) est considérée comme une avancée majeure dans le traitement automatique des langues, en particulier pour les tâches de classification ou d'étiquetage de séquences traitées par des approches d'apprentissage supervisé. Cependant, ce type de plongements représente également un changement significatif du point de vue de la sémantique distributionnelle. Alors que les approches

Volume 1 : conférence principale, pages 26–36.

antérieures, y compris les plongements statiques de mots tels que les modèles Skip-gram ou CBOW (Mikolov *et al.*, 2013), construisaient la représentation distributionnelle d'un mot en cumulant ses contextes d'occurrence, les modèles de langage neuronaux tels qu'ELMo ou BERT produisent une représentation pour chaque occurrence des mots, ce qui, à la fois, pose des problèmes pour l'étude des propriétés sémantiques de ces représentations mais ouvrent aussi certaines opportunités.

Comme l'illustrent les travaux de Rogers *et al.* (2020), les propriétés des plongements contextuels de mots sont au centre de nombreuses études récentes, notamment dans le cas de BERT. Rogers *et al.* (2020) signalent principalement les travaux portant sur la sémantique au niveau phrastique, soit dans le contexte de l'étiquetage en rôles sémantiques (Tenney *et al.*, 2019), soit dans des tâches relevant de la substitution lexicale (Ettinger, 2020; Garí Soler *et al.*, 2019). Mais les travaux récents de Vulić *et al.* (2020) présentent clairement le plus grand ensemble d'expériences concernant les propriétés sémantiques des modèles BERT et RoBERTa, y compris concernant leur corrélation avec des jugements humains en termes de similarité sémantique. D'autres études ont également examiné les plongements contextuels d'un point de vue sémantique mais dans un contexte plus spécifique : l'impact des fonctions objectifs utilisées pour l'entraînement de ces modèles (Mickus *et al.*, 2020), leur niveau de contextualisation (Ethayarajh, 2019), leurs biais possibles (Bommasani *et al.*, 2020), leur capacité à représenter les sens des mots (Coenen *et al.*, 2019; Chronis & Erk, 2020) ou à construire des représentations pour des mots rares (Schick & Schütze, 2020).

Bien que notre travail ait des liens avec certains de ces travaux, son étude des plongements contextuels repose sur une méthode spécifique, fondée sur des principes distributionnels, qui vise à caractériser les plongements contextuels en termes de relations paradigmatiques à la fois au niveau des occurrences de mots et des mots. Plus précisément, nous présentons les contributions suivantes :

— nous montrons que l'hypothèse distributionnelle peut être appliquée au niveau des occurrences de mots avec des plongements contextuels et que plus spécifiquement, ELMo et BERT présentent des propriétés sémantiques propres relevant davantage de la similarité sémantique que de la proximité sémantique ;

— nous montrons comment la transposition de cette méthode au niveau des mots peut servir à étudier les différences entre les plongements contextuels et les plongements statiques de mots.

2 Méthode de test de plongements contextuels

2.1 Principes

Avec les modèles de plongements contextuels, la similarité entre les plongements de mots ne peut être calculée que pour les mots en contexte, c'est-à-dire les occurrences de mots, ce qui rend l'application des principes distributionnels au niveau des mots plus difficile mais offre la possibilité de les appliquer au niveau de ces occurrences. Plus précisément, selon Harris (1954), l'hypothèse distributionnelle se définit par :

« [...] difference of meaning correlates with difference of distribution. »

ce qui, exprimé de façon plus développée, donne :

« If A and B have almost identical environments except chiefly for sentences which contain both, we say they are synonyms : *oculist* and *eye-doctor*. If A and B have some environments in common and some not (e.g.

oculist and *lawyer*) we say that they have different meanings, the amount of meaning difference corresponding roughly to the amount of difference in their environments. »

Ce principe conduit à l'idée de *substituabilité*, que l'on retrouve dans :

« We group A and B into a substitution set whenever A and B each have the same (or partially same) environments X [...] »

Classiquement, les environnements mentionnés par Harris (1954) font référence aux ensembles de cooccurrents de *A* et *B*. Dans le cas des approches prédictives (Baroni *et al.*, 2014), ces environnements sont condensés dans des représentations distribuées appelées plongements de mots. Ainsi, selon les principes ci-dessus, plus deux plongements sont proches l'un de l'autre, plus les mots qui leur sont associés sont substituables et donc, plus ces mots sont susceptibles d'entretenir une relation sémantique proche de la synonymie. Dans le cas des modèles contextuels, le plongement construit pour une occurrence d'un mot avec ce type de modèles intègre deux dimensions : l'une, résultant de l'entraînement sur la tâche de modélisation linguistique, prend en compte l'agrégation de tous les contextes des occurrences du mot, comme pour tous les modèles distributionnels ; la seconde caractérise le contexte local de l'occurrence considérée, c'est-à-dire les mots qui l'entourent.

Dans notre cas, nous nous concentrons sur la première dimension pour évaluer la similarité entre les mots, ce qui nécessite de pouvoir contrôler la seconde. Nous proposons d'effectuer ce contrôle très simplement en nous appuyant sur le principe de base de l'approche distributionnelle : pour évaluer la similarité de deux mots, nous plaçons les deux mots dans le même contexte, c'est-à-dire à la même position dans la même phrase, et calculons un plongement de mot pour chacun d'eux en appliquant le modèle de langage contextuel considéré. En pratique, ce contrôle est mis en œuvre par une substitution : pour deux mots w_1 et w_2, une phrase S_i contenant une occurrence de w_1 est sélectionnée et une représentation contextualisée de w_1 est construite ; w_1 est ensuite remplacé par w_2 dans la phrase S_i et une représentation contextualisée de w_2 est construite de la même manière que pour w_1. En calculant la similarité entre les représentations de w_1 et w_2, nous pouvons évaluer dans quelle mesure w_1 peut être remplacé par w_2 et, selon le principe de substituabilité mentionné ci-dessus, obtenir ainsi une évaluation de leur similarité sémantique.

Cette façon de faire peut d'une certaine façon être vue comme un renversement de la substitution lexicale (McCarthy & Navigli, 2009) : au lieu de choisir un substitut à partir de sa similarité avec un contexte et un mot de référence, un substitut donné permet d'évaluer la similarité avec un mot de référence. Cette évaluation est bien sûr limitée à la phrase sélectionnée mais l'application de cette stratégie à un ensemble de phrases donne une image plus globale de la relation sémantique entre les deux mots. Il est à noter que w_1 et w_2 n'ont pas de rôles symétriques : nous évaluons la similarité de w_2 par rapport à w_1 puisque la représentation de w_2 est construite dans le contexte d'une phrase dans laquelle w_1 apparaît initialement. D'un point de vue pratique, les représentations contextuelles de w_1 et w_2 sont obtenues en encodant la phrase qui les abrite avec le modèle visé et en les extrayant des couches internes de ce modèle, avec donc une représentation par couche. Dans la suite de l'article, les termes *mot source*, *mot cible* et *phrase test* feront référence respectivement à w_1, w_2 et S_i.

2.2 Étude sémantique des plongements contextuels

La section précédente donne un principe pour évaluer la similarité d'un mot cible vis-à-vis d'une occurrence d'un mot source dans le contexte d'une phrase. Nous montrons maintenant comment ce principe peut être appliqué pour explorer les relations sémantiques sous-jacentes à des modèles

contextuels tels qu'ELMo ou BERT. Plus précisément, l'idée est, pour chaque mot d'un ensemble de mots sources, de rassembler un ensemble de mots cibles de telle sorte que chaque paire (mot source, mot cible) soit liée par une relation sémantique de type connu. À partir d'une phrase test contenant une occurrence du mot source, ses mots cibles peuvent être classés en fonction de leur valeur de similarité avec le mot source selon le principe de substituabilité décrit précédemment. Le classement qui en résulte ordonne indirectement les types de relations sémantiques associées aux mots cibles et donne ainsi des indications sur les propriétés sémantiques d'ELMo ou de BERT.

Dans ce travail, nous nous intéressons plus particulièrement aux relations paradigmatiques, plus précisément la synonymie [SYN], l'hyperonymie [HYPE], l'hyponymie [HYPO] et la cohyponymie [COHYP], et nous adoptons WordNet 3.0 comme ressource de référence pour ces types de relations. Cependant, WordNet (Miller, 1990) est fondé sur la notion de synset alors que les phrases contiennent des mots et non des sens de mots. Pour contourner cette difficulté, nous utilisons comme phrases test des phrases du corpus SemCor (Miller *et al.*, 1993), un sous-ensemble du Corpus Brown dont les mots de classe ouverte sont étiquetés avec des synsets de WordNet. Ainsi, le sens d'une occurrence d'un mot source est connu et la relation avec le mot cible dépend de ce sens, ce qui rend l'utilisation de la substitution particulièrement précise. Par exemple, le deuxième sens du mot source *disaster* (*an event resulting in great loss and misfortune*) a, parmi d'autres, la phrase test suivante dans le corpus SemCor :

[1] Since the 1946 **disaster** there have been 15 tsunami in the Pacific, but only one was of any consequence.

Ce sens du mot *disaster* a des mots tels que *cataclysm* ou *catastrophe* comme synonymes, *misfortune* comme hyperonyme, *tsunami* or *meltdown* comme hyponymes et *adversity* or *misadventure* comme cohyponymes. Pour évaluer dans quelle mesure ELMo, par exemple, est plus orienté vers la synonymie que l'hyperonymie, la phrase test [1] est transformée pour donner les phrases [2] et [3] en remplaçant le mot source par un synonyme ou un hyperonyme.

[2] Since the 1946 **catastrophe** there have been 15 tsunami in the Pacific, but only one was of any consequence.
[3] Since the 1946 **misfortune** there have been 15 tsunami in the Pacific, but only one was of any consequence.

Les trois phrases sont encodées à l'aide d'ELMo et la représentation du mot source dans la phrase [1] et des deux mots cibles dans les phrases [2] et [3] sont extraites des couches internes d'ELMo. Comme ELMo comporte trois couches – une couche non contextuelle d'entrée, la couche 0, et deux couches contextuelles, les couches 1 et 2 – nous obtenons trois représentations pour chaque mot. Dans cet exemple et pour la couche 1, la similarité, évaluée classiquement par la mesure *cosinus* appliquée entre la représentation du mot source et celle de la cible synonyme *catastrophe* est égale à 0,89 alors que la similarité du mot source et de la cible hyperonyme *misfortune* n'est égale qu'à 0,55, ce qui donne dans ce cas un net avantage à la synonymie par rapport l'hyperonymie. Le processus est le même pour BERT, sauf que nous avons 12 couches (de la couche 1 à la couche 12) dans ce cas. Une image globale des propriétés sémantiques d'ELMo ou de BERT est obtenue en considérant un nombre significatif de mots sources et de mots cibles dans le contexte d'un grand nombre de phrases test du corpus SemCor. De plus, nous enrichissons cette image en considérant les relations entre les plongements d'ELMo et de BERT et les plongements de mots statiques plus classiques en ajoutant comme cibles DIST_NGH, les voisins les plus similaires aux mots cibles de notre étude obtenus grâce à la mesure *cosinus* sur la base d'un modèle Skip-gram entraîné à partir d'un grand corpus.

Cette étude des propriétés sémantiques d'ELMo et de BERT peut être considérée comme une sorte de sondage sémantique et être liée en tant que telle aux travaux de Schick & Schütze (2020) et à

leur méthode *WordNet Language Model Probing*. Néanmoins, leur objectif global est très différent du nôtre et leur tâche de sondage, fondée sur une notion de patron, est plus adaptée aux relations syntagmatiques qu'aux relations paradigmatiques.

2.3 Plongements contextuels versus plongements statiques

Outre l'étude des propriétés sémantiques des plongements contextuels, la méthode que nous avons présentée peut également être adaptée pour étudier les différences entre ces plongements et les plongements statiques du point de vue de ces mêmes propriétés. Plus précisément, l'idée est de définir les cibles en remplaçant les relations sémantiques de WordNet par des relations caractérisant les plongements statiques. En raison de la nature distributionnelle de ces plongements, nous avons choisi d'associer comme cibles à chaque clé, correspondant à un mot, ses voisins distributionnels les plus similaires selon les plongements statiques considérés. Comme précédemment, les cibles sont ordonnées en fonction d'un ensemble de phrases test par les plongements contextuels étudiés et nous utilisons les relations paradigmatiques de WordNet pour déterminer a posteriori les types de relations que les plongements contextuels favorisent par rapport aux plongements statiques. Puisque les clés sont des mots et non des sens de mots dans ce protocole, nous agrégeons les représentations contextuelles des clés et des cibles construites à partir des phrases test en les moyennant, comme recommandé par Bommasani *et al.* (2020), pour obtenir des représentations au niveau des mots.

3 Expérimentations

3.1 Étude sémantique des plongements contextuels

3.1.1 Cadre d'expérimentation

La mise en œuvre des principes présentés précédemment est associée à certains choix concernant les phrases test et les relations sémantiques entre les mots sources et les mots cibles. Notre premier choix a été de nous concentrer sur les noms. Le nombre de mots cibles pour chaque mot source a été fixé à 40 pour avoir des résultats comparables pour tous les mots sources. De plus, nous n'avons retenu que les mots sources ayant des mots cibles pour les cinq types de relations que nous considérons, une fois encore pour l'homogénéité des résultats entre mots sources. Nous avons également limité le nombre de mots cibles pour chaque type de relations à 10, avec une limite supplémentaire à 30 pour l'ensemble des mots cibles issus de relations de WordNet. En pratique, cette limite ne concerne que les relations d'hyponymie et de cohyponymie, dont le nombre tend à être élevé. La première colonne du tableau 1 donne le nombre moyen de mots cibles de chaque mot source en fonction de leur type de relations sémantiques.

Par ailleurs, nous avons adopté les définitions suivantes pour chaque type de cibles issues de relations de WordNet :

- synonymes [SYN] : tous les mots de $Synset_{srce}$, le synset correspondant au sens du mot source présent dans la phrase test considérée ;
- hyperonymes [HYPE] : tous les mots des synsets $Synset_{hype}$ ayant un lien direct d'hyperonymie avec $Synset_{srce}$;

	#cibles	aléatoire	ELMo			BERT					
			L_0	L_1	L_2	L_1	L_3	L_5	L_8	L_{10}	L_{12}
SYN	2,1	7,4	30,9	29,3	26,9	33,5	34,5	36,1	**36,7**	36,0	35,1
HYPE	1,9	6,7	4,3	6,1	6,2	7,8	9,3	9,9	10,9	11,1	**11,2**
HYPO	5,9	20,9	11,4	13,8	**14,8**	10,5	11,1	11,7	12,3	11,9	12,3
COHYP	8,3	29,4	6,7	7,9	**9,6**	6,4	6,7	7,5	7,7	7,8	7,9
DIST_NGH	10	35,5	**46,6**	42,9	42,5	41,7	38,4	34,7	32,3	33,1	33,5

TABLE 1 – P@1 ($\times$100) pour l'ordonnancement, fondé sur les phrases du corpus SemCor, des mots cibles associés à un ensemble de mots sources

— hyponymes [HYPO] : tous les mots des synsets ayant une relation directe d'hyponymie avec $Synset_{srce}$;
— cohyponymes [COHYP] : tous les mots des synsets, à l'exception de $Synset_{srce}$, ayant une relation directe d'hyponymie avec les synsets $Synset_{hype}$.

Comme mentionné précédemment, les mots cibles DIST_NGH sont obtenus par un modèle Skip-gram, entraîné sur un sous-ensemble d'1 milliard de mots du corpus anglais annoté Gigaword (Napoles *et al.*, 2012) avec les meilleures valeurs d'hyperparamètres de (Baroni *et al.*, 2014) [1]. Plus précisément, nous utilisons ce modèle pour sélectionner les 10 premiers voisins distributionnels de chaque mot source, parmi un vocabulaire de 20 813 noms, ne correspondant pas à un mot cible issu d'une relation de WordNet. Dans ce cas, cette sélection n'est pas faite selon le sens du mot cible dans une phrase test puisque nous n'avons pas accès aux sens des mots.

En ce qui concerne les phrases test, une limite supérieure de 20 phrases a également été fixée pour chaque mot source, toujours pour avoir des résultats comparables entre mots sources. Finalement, notre évaluation s'appuie sur 41 079 phrases du corpus SemCor, ce qui représente environ 4,5 phrases par sens en moyenne et 7,9 phrases pour chacun des 5 241 mots sources. Ces derniers couvrent un large spectre de fréquences puisque si nous nous référons à la sous-partie que nous utilisons du corpus Gigaword, le mot source le plus fréquent, *year*, a 2 991 899 occurrences alors que les mots sources les moins fréquents, comme *inadvertence*, n'ont que 22 occurrences.

3.1.2 Évaluation

Le résultat du classement des mots cibles sélectionnés pour tous nos mots sources et toutes nos phrases test est présenté dans le tableau 1 pour différentes couches d'ELMo et de BERT (L_x) et chaque type de mots cibles. De plus, la deuxième colonne fournit les valeurs de précision au rang 1 (P@1) d'un classement aléatoire selon la distribution de la première colonne. P@1 dans ce contexte correspond à la proportion de phrases test qui classent en premier un mot cible lié à un mot source avec un type de relations spécifique (un par ligne).

Ce tableau montre en premier lieu que les différentes couches d'ELMo et de BERT n'ont pas exactement les mêmes propriétés sémantiques, ce qui confirme les conclusions d'Ethayarajh (2019) ou de Wu *et al.* (2020). Il montre également certaines similitudes et différences entre ELMo et BERT. La différence la plus évidente est que BERT obtient des résultats beaucoup plus élevés

1. Fréquence minimale=5, taille vecteurs=300, taille fenêtre=5, 10 exemples négatifs et 10^{-5} pour le sous-échantillonnage des mots les plus fréquents.

qu'ELMo pour les synonymes et les hyperonymes, mais des résultats un peu plus faibles pour les hyponymes et les cohyponymes. BERT favorise donc la similarité sémantique plutôt que la proximité sémantique au sens de Budanitsky & Hirst (2006). On peut en effet considérer que les hyponymes et les cohyponymes, malgré leur nature paradigmatique, sont moins représentatifs, en termes de similarité sémantique, que les synonymes et les hyperonymes. La tendance est encore plus évidente pour les relations DIST_NGH provenant des plongements statiques, qui sont davantage susceptibles d'être de nature syntagmatique.

ELMo et BERT présentent également quelques différences quant aux propriétés de leurs couches respectives. Alors que la capacité d'ELMo à classer au rang 1 les synonymes diminue régulièrement à mesure que l'on considère des couches de plus en plus hautes, elle augmente d'abord jusqu'à la couche 8 pour BERT, puis tend à diminuer. Cette observation doit également être mise en relation avec les résultats d'Ethayarajh (2019), qui a observé que les représentations des mots sont plus contextualisées à mesure que le niveau de leurs couches augmente. Du point de vue d'ELMo, cette plus grande contextualisation des représentations tendrait donc à favoriser la proximité sémantique au détriment de la similarité sémantique. Ce n'est pas complètement surprenant dans la mesure où, du point de vue sémantique, la contextualisation est plus susceptible de conduire à des relations syntagmatiques que paradigmatiques. Ce résultat est moins évident pour BERT puisque ses premières couches ont la meilleure précision pour les relations DIST_NGH mais un changement plus compatible avec les résultats d'ELMo se produit après la couche 8 concernant le classement des synonymes et des relations DIST_NGH.

Cependant, ELMo et BERT ont également de fortes convergences. En premier lieu, l'effet le plus significatif du classement des mots cibles par ELMo et BERT est obtenu pour les synonymes. P@1 est jusqu'à quatre fois plus élevée pour la meilleure couche d'ELMO et cinq fois pour la meilleure couche de BERT par rapport au classement aléatoire initial. L'application de l'hypothèse distributionnelle telle que décrite à la section 2.1 est donc une méthode intéressante pour identifier les synonymes d'un mot parmi une liste de ses voisins distributionnels. Plus globalement, si certaines différences peuvent être notées entre les couches de ces modèles en termes d'orientation sémantique, elles ont toutes un fort penchant pour la synonymie. Nous pouvons également observer que tant ELMo que BERT ont une forte corrélation inverse entre leur P@1 pour les synonymes et leur P@1 pour les relations DIST_NGH : lorsqu'une couche tend à favoriser une stricte similarité sémantique, elle obtient logiquement des résultats moins bons pour la proximité sémantique. Cependant, cette corrélation ne conduit pas à des résultats pour les relations DIST_NGH beaucoup plus faibles que les résultats d'un classement aléatoire, en particulier pour ELMo, ce qui suggère qu'ELMo et BERT ne sont pas radicalement différents des plongements statiques du point de vue de la similarité sémantique qu'ils véhiculent.

Un examen plus approfondi des mot cibles DIST_NGH classés en premier montre également qu'ils ont souvent une forte relation sémantique avec le mot source. Dans certains cas, il est l'un de ses synonymes mais pour un sens différent qui est proche du sens de l'occurrence courante. Par exemple, dans la phrase :

Land reform programs need to be supplemented with programs for promoting rural credits [...] in **agriculture**.

le mot classé en premier par la première couche d'ELMo, *farming*, est synonyme du deuxième sens de *agriculture – the practice of cultivating the land or raising stock* – alors que cette occurrence est étiquetée dans le SemCor avec son premier sens – *a large-scale farming enterprise*. Dans d'autres cas, le mot cible de rang 1 est en fait un synonyme du mot source mais n'est pas considéré comme tel dans WordNet. Par exemple, le mot cible *capability* pour le mot source *ability*. Enfin, il est également

	#relations		ELMo			BERT					
	référence	Skip-gram	L_0	L_1	L_2	L_1	L_3	L_5	L_8	L_{10}	L_{12}
SYN	3,5	18,6	22,6	**23,0**	21,4	20,1	20,6	20,9	21,4	21,3	21,9
HYPE	5,0	6,9	5,8	6,6	6,3	8,1	8,3	8,6	8,9	**9,0**	8,5
HYPO	10,8	11,8	13,3	**14,2**	13,7	10,5	10,8	11,3	11,5	10,9	10,2
COHYP	56,3	27,9	33,5	**34,8**	33,0	30,0	30,7	31,2	31,3	31,2	31,6

TABLE 2 – P@1 ($\times 100$) pour l'ordonnancement des voisins distributionnels du modèle Skip-gram par les plongements contextuels

fréquent de trouver comme mot cible de rang 1 un antonyme du mot source, comme *inaction* pour le mot source *action*. Bien que ce ne soit pas un résultat attendu du point de vue de notre référence, les antonymes sont connus pour être très similaires aux synonymes sur le plan distributionnel et leur présence confirme donc la tendance observée à favoriser la similarité sémantique.

3.2 Plongements contextuels versus plongements statiques

3.2.1 Cadre d'expérimentation

La principale différence avec l'expérience précédente concerne les phrases test. Elles sont ici sélectionnées aléatoirement dans une sous-partie du corpus utilisé pour l'entraînement des plongements statiques et les clés n'y sont pas sémantiquement désambiguïsées. Plus précisément, nous avons sélectionné 10 phrases test pour chaque clé, d'une taille comprise entre 10 et 90 mots pour avoir un contexte significatif mais ciblé. Pour les plongements statiques, nous nous appuyons sur le même modèle Skip-gram que celui utilisé pour extraire les relations DIST_NGH. Le processus d'ordonnancement est appliqué aux mêmes 5 241 clés que dans la première expérience et se concentre sur leurs 10 premiers voisins distributionnels, résultant de l'application de la mesure cosinus. Comme dans (Piasecki *et al.*, 2018), nos voisins de référence sont obtenus en extrayant de WordNet les mots liés à une clé par les mêmes types de relations qu'à la section 3.1. Cependant, le nombre de relations est ici plus important puisque nous considérons tous les synsets dans lesquels la clé est présente du fait de l'absence de désambiguïsation sémantique des clés. La mesure utilisée est comme précédemment la précision au premier rang (P@1) des voisins reclassés pour chaque type de relations.

3.2.2 Évaluation

Le tableau 2 compare les voisins distributionnels ordonnés par le modèle Skip-gram avec leur ordonnancement par ELMo et BERT selon la méthodologie que nous proposons. La deuxième colonne de ce tableau donne le nombre moyen de relations du type considéré par clé dans WordNet, c'est-à-dire la richesse de la référence pour l'évaluation.

La première observation est que si ces plongements contextuels favorisent significativement [2] les synonymes et les cohyponymes par rapport à Skip-gram, la situation est plus complexe pour les hyperonymes et les hyponymes : ELMo dégrade les résultats de Skip-gram pour les hyperonymes mais les améliore pour les hyponymes alors que BERT fait exactement le contraire.

2. Les différences sont jugées significatives selon un test de Wilcoxon apparié si $p \leq 0,01$.

On peut également noter que le modèle Skip-gram favorise largement la cohyponymie par rapport aux autres relations lexicales, une tendance qui se trouve globalement accentuée par ELMo et BERT, avec une plus grande amplitude dans le cas d'ELMo. De ce point de vue, il est intéressant de noter que dans le tableau 1, l'ordonnancement des cohyponymes par ELMo et BERT est au contraire très nettement inférieur à celui obtenu par un ordonnancement aléatoire. Cette différence entre les deux expériences illustre la nature contextuelle de ces modèles mais aussi la limite de cette contextualisation. Dans notre première expérience, les clés sont sémantiquement désambiguïsées dans les phrases test et les cibles sont liées au sens de ces clés désambiguïsées. Dans une telle situation, un modèle contextuel peut favoriser les cibles les plus liées sémantiquement à leur clé, comme les synonymes, car il produit des représentations spécifiques au contexte considéré, qui est lié au sens de la clé. Au contraire, dans notre seconde expérience, les cibles couvrent tous les sens de la clé et dans une telle configuration, les plongements contextuels favorisent la proximité sémantique plutôt que la similarité sémantique et sont ainsi plus proches des plongements statiques, même s'ils tendent globalement à les surpasser.

Cet effet a un impact beaucoup plus important pour les synonymes dans le cas de BERT que pour ELMo, ce qui résulte probablement d'une plus grande sensibilité de BERT au contexte qu'ELMo. Cependant, il ne modifie pas la configuration des résultats entre les différentes couches de BERT, avec de meilleurs résultats autour de la couche 8, alors que les meilleurs résultats d'ELMo sont assez étonnamment obtenus par une couche plus contextualisée que précédemment.

4 Conclusion et perspectives

Dans cet article, nous avons étudié comment le principe distributionnel de substitution peut être utilisé comme une forme de sonde pour tester le type de similarité sémantique véhiculé par ELMo et BERT. Des expériences menées avec les relations paradigmatiques de WordNet et le corpus SemCor au niveau des sens des mots ont montré que la nature contextuelle de ces modèles favorise clairement la synonymie en termes de relations lexicales. Dans un second temps, nous avons adapté la même méthode pour étudier les différences entre les plongements contextuels et statiques, avec la conclusion que le biais des plongements contextuels en faveur de la similarité sémantique observé au niveau des occurrences de mots est fortement réduit au niveau des mots hors contexte par rapport aux plongements statiques.

Une des extensions les plus directes de ce travail est l'élargissement de la méthode de comparaison des plongements contextuels et statiques au problème de l'amélioration des thésaurus distributionnels par le biais du réordonnancement de leurs voisins. L'ordonnancement des voisins distributionnels n'est vu dans le travail présenté que comme une façon de caractériser les différences sémantiques entre plongements contextuels et statiques. Mais il peut aussi être envisagé sous l'angle d'une amélioration des thésaurus distributionnels construits à partir de plongements statiques en mettant en avant les voisins distributionnels les plus sémantiquement liés à leurs entrées. Dans cette optique, une étude des meilleures options pour l'agrégation au niveau des mots hors contexte des informations fournies au niveau des occurrences de mots par les plongements contextuels serait d'un grand intérêt.

Remerciements

Le travail présenté dans cet article a été réalisé dans le cadre du projet ADDICTE [3] (Analyse distributionnelle en domaine spécialisé), financé par l'Agence Nationale de la Recherche (ANR-17-CE23-0001).

Références

BARONI M., DINU G. & KRUSZEWSKI G. (2014). Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, p. 238–247, Baltimore, Maryland.

BOMMASANI R., DAVIS K. & CARDIE C. (2020). Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, p. 4758–4781, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.431.

BUDANITSKY A. & HIRST G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, **32**(1), 13–47.

CHRONIS G. & ERK K. (2020). When is a bishop not like a rook ? When it's like a rabbi ! Multi-prototype BERT embeddings for estimating semantic relationships. In *24th Conference on Computational Natural Language Learning (CoNLL 2020)*, p. 227–244, Online : Association for Computational Linguistics.

COENEN A., REIF E., YUAN A., KIM B., PEARCE A., VIÉGAS F. & WATTENBERG M. (2019). Visualizing and Measuring the Geometry of BERT. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ BUC, E. FOX & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 32, p. 8594–8603 : Curran Associates, Inc.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2019)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

ETHAYARAJH K. (2019). How Contextual Are Contextualized Word Representations ? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, p. 55–65, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1006.

ETTINGER A. (2020). What BERT Is Not : Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, **8**, 34–48. DOI : 10.1162/tacl_a_00298.

GARÍ SOLER A., APIDIANAKI M. & ALLAUZEN A. (2019). Word Usage Similarity Estimation with Sentence Representations and Automatic Substitutes. In *Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, p. 9–21, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/S19-1002.

3. https://anr-addicte.ls2n.fr/

HARRIS Z. S. (1954). Distributional Structure. *Word*, **10**(2-3), 146–162.

MCCARTHY D. & NAVIGLI R. (2009). The English lexical substitution task. *Language resources and evaluation*, **43**(2), 139–159.

MICKUS T., CONSTANT M., PAPERNO D. & VAN DEEMTER K. (2020). What do you mean, BERT ? Assessing BERT as a Distributional Semantics Model. *Society for Computation in Linguistics*, **3**. DOI : 10.7275/t778-ja71.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

MILLER G. A. (1990). WordNet : An On-Line Lexical Database. *International Journal of Lexicography*, **3**(4).

MILLER G. A., LEACOCK C., TENGI R. & BUNKER R. T. (1993). A Semantic Concordance. In *Human Language Technology*, Plainsboro, USA.

NAPOLES C., GORMLEY M. R. & VAN DURME B. (2012). Annotated Gigaword. In *NAACL Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, p. 95–100, Montréal, Canada.

PETERS M., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep Contextualized Word Representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2018)*, p. 2227–2237, New Orleans, Louisiana, USA : Association for Computational Linguistics.

PIASECKI M., CZACHOR G., JANZ A., KASZEWSKI D. & KEDZIA P. (2018). Wordnet-based evaluation of large distributional models for polish. In *9th Global WordNet Conference (GWC 2018)*.

ROGERS A., KOVALEVA O. & RUMSHISKY A. (2020). A Primer in BERTology : What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, **8**, 842–866. DOI : 10.1162/tacl_a_00349.

SCHICK T. & SCHÜTZE H. (2020). Rare Words : A Major Problem for Contextualized Embeddings and How to Fix It by Attentive Mimicking. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, p. 8766–8774, New York, USA.

TENNEY I., XIA P., CHEN B., WANG A., POLIAK A., MCCOY R. T., KIM N., DURME B. V., BOWMAN S., DAS D. & PAVLICK E. (2019). What do you learn from context ? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

VULIĆ I., PONTI E. M., LITSCHKO R., GLAVAŠ G. & KORHONEN A. (2020). Probing Pretrained Language Models for Lexical Semantics. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, p. 7222–7240, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.586.

WU J., BELINKOV Y., SAJJAD H., DURRANI N., DALVI F. & GLASS J. (2020). Similarity Analysis of Contextual Word Representation Models. In *58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, p. 4638–4655, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.422.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

La génération de textes artificiels en substitution ou en complément de données d'apprentissage

Vincent Claveau[1] Antoine Chaffin[1,2] Ewa Kijak[1]
(1) IRISA - CNRS, Univ. Rennes, Campus de Beaulieu, 35042 Rennes, France
(2) IMATAG, Rennes, France
{vincent.claveau,ewa.kijak}@irisa.fr, antoine.chaffin@imatag.com

RÉSUMÉ

La qualité des textes générés artificiellement s'est considérablement améliorée avec l'apparition des *transformers*. La question d'utiliser ces modèles pour augmenter les données d'apprentissage pour des tâches d'apprentissage supervisé se pose naturellement. Dans cet article, cette question est explorée sous 3 aspects : (i) les données artificielles sont-elles un complément efficace ? (ii) peuvent-elles remplacer les données d'origines quand ces dernières ne peuvent pas être distribuées, par exemple pour des raisons de confidentialité ? (iii) peuvent-elles améliorer l'explicabilité des classifieurs ? Différentes expériences sont menées sur une tâche de classification en utilisant des données générées artificiellement en adaptant des modèles GPT-2. Les résultats montrent que les données artificielles ne sont pas encore suffisamment bonnes et nécessitent un pré-traitement pour améliorer significativement les performances. Nous montrons que les approches sac-de-mots bénéficient le plus de telles augmentations de données.

ABSTRACT

Generating artificial texts as substitution or complement of training data

The quality of artificially generated texts has considerably improved with the advent of transformers. The question of using these models to generate learning data for supervised learning tasks naturally arises. In this article, this question is explored under 3 aspects: (i) are artificial data an efficient complement? (ii) can they replace the original data when those are not available or cannot be distributed for confidentiality reasons? (iii) can they improve the explainability of classifiers? Different experiments are carried out on a classification task using artificially generated data by fine-tuned GPT-2 models. The results show that such artificial data are not yet good enough and require pre-processing to significantly improve performance. We show that bag-of-word approaches benefit the most from such data augmentation.

MOTS-CLÉS : Génération de textes, augmentation de données, classification.

KEYWORDS: Text generation, data augmentation, classification.

1 Introduction

Si la génération artificielle de texte n'est pas une tâche nouvelle, les approches récentes à base de *transformers* offrent des performances suffisamment bonnes pour être employées dans de nombreux contextes (Vaswani *et al.*, 2017). Dans cet article, nous explorons l'utilisation de données générées pour des tâches d'apprentissage supervisé dans différents contextes d'utilisation afin de compléter

les données d'entraînement originales (pour obtenir de meilleures performances) ou de se substituer (intégralement) aux données originales (par exemple, quand ces dernières ne peuvent pas être distribuées pour des raisons de confidentialité (Amin-Nejad *et al.*, 2020)). La génération de ces données est faite avec un modèle de langue neuronal appris sur les données d'origine.

Précisément, les principales questions de recherche abordées dans cet article sont les suivantes :

1. quel est l'intérêt de la génération pour améliorer les performances des approches à base d'apprentissage (complément) ;
2. quel est l'intérêt de la génération pour remplacer les données d'origines (substitution) ;
3. quel est l'intérêt de la génération pour des classifieurs neuronaux et ceux dits "explicables" reposant sur des représentations sac-de-mots.

Dans la suite de cet article, après une présentation des travaux connexes (sec. 2), nous détaillons le processus d'augmentation de données que nous mettons en œuvre en section 3. Les données expérimentales sont décrites en section 4. Les expériences et résultats pour nos différentes questions de recherche sont détaillés en section 5 pour les modèles neuronaux et section 6 pour les modèles expploitant des représentations sac-de-mots.

2 Travaux connexes

L'augmentation de données pour des tâches de TAL a déjà été explorée dans plusieurs travaux. Certains proposent des modifications automatiques plus ou moins complexes des exemples originaux afin de créer une version différente en surface mais identique au sens de la tâche (même classe, même relation entre les mots...) en remplaçant par exemple des mots par leur synonymes (Kobayashi, 2018; Wei & Zou, 2019; Mueller & Thyagarajan, 2016; Jungiewicz & Smywinski-Pohl, 2019). Les synonymes sont alors tirés de ressources langagières tels WordNet, de thésaurus distributionnels ou de plongements de mots (statiques).

Dans une veine similaire, puisque ne modifiant que localement les données originales, il existe des approches neuronales exploitant les modèles de langue à base de masque de type BERT et donc des plongements contextuels. Celles-ci fonctionnent en conditionnant le remplacement du jeton [MASK] par un mot en fonction de la classe attendue (Wu *et al.*, 2019). Cela produit un nouvel exemple avec un remplacement d'un mot par un mot sémantiquement proche (idéalement un synonyme), mais ce nouvel exemple n'est pas totalement différent (la structure du nouvel exemple est très similaire à l'exemple original) comme nous proposons de le faire.

D'autres approches exploitent les capacités des modèles de langue tels GPT-2 (Generative Pre-Trained Transformers) afin de produire des données proches de la distribution du jeu initial en grande quantité. En recherche d'information, ce principe a par exemple été utilisé pour augmenter les requêtes (Claveau, 2020b). Plus proche encore, la génération est exploitée pour l'extraction de relations (Papanikolaou & Pierleoni, 2020), la classification de sentiments de critiques et de questions (Kumar *et al.*, 2020) ou la prédiction de réadmission et la classification phénotypique (Amin-Nejad *et al.*, 2020). Notre article s'inscrit dans cette lignée de travaux. Notre intérêt est ici d'examiner les gains et pertes des différents scénarios d'emploi des données artificielles, de leur préparation, et d'examiner leurs effets sur différentes familles de classifieurs.

3 Génération de données artificielles

On suppose disposer d'un ensemble de textes (originaux) $\mathcal{T}$ divisé en n classes c_i, à partir desquels on souhaite générer des textes artificiels $\mathcal{G}_{c_i}$ pour chaque classe c_i. Nous employons les modèles GPT pour générer les textes artificiels. Ces modèles sont construits en empilant des *transformers* (plus précisément des décodeurs), entraînés sur de grands corpus par auto-régression, c'est-à-dire sur une tâche de prédiction du mot (ou *token*) suivant, sachant les précédents. La seconde version, GPT-2 (Radford *et al.*, 2019), contient 1,5G de paramètres pour son plus gros modèle, entraînés sur plus de 8 millions de documents issus de Reddit (i.e. du langage général comme des discussions sur des articles de presse, principalement en anglais).

3.1 Adaptation du modèle de langue.

Pour cette étape d'adaptation fine (*fine-tuning*), on part du modèle moyen (774M de paramètres) pré-entraîné pour l'anglais et mis à disposition par OpenAI[1].

Dans les travaux présentés dans cet article, nous adaptons un modèle de langue par classe. Une autre manière d'entraîner disponible dans la littérature consiste à adapter un unique modèle, mais de le conditionner par un *token* spécial indiquant la classe attendue en début de séquence. Du fait du peu de données disponibles par classe vis-à-vis du nombre de paramètres du modèle GPT-2, il est important de contrôler l'adaptation pour éviter le sur-apprentissage. Pour cela, nous limitons le nombre d'*epochs* à 2 000 ; les autres paramètres d'adaptation sont ceux par défaut. Sur une carte GPU Tesla V100, cette étape de *fine-tuning* dure environ 1h.

3.2 Génération.

Pour chacune des classes c_i du jeu de données $\mathcal{T}$, nous utilisons le modèle correspondant pour générer des textes artificiels $\mathcal{G}_{c_i}$ qui, nous l'espérons, relèveront bien de la même classe. Nous fournissons des amorces pour ces textes sous la forme d'une balise de début de texte suivi d'un mot tiré aléatoirement dans l'ensemble des textes originaux. Plusieurs paramètres peuvent influer sur la génération. Nous avons laissé les valeurs usuelles que nous redonnons ici, sans les détailler (voir la documentation GPT-2), à des fins de reproductibilité : `temp.` = 0,7, `top_p` = 0,9, `top_k` = 40.

Les textes générés pour la classe c_i contenant une séquence de 5 mots consécutifs apparaissant identiquement dans un texte de $\mathcal{T}_{c_i}$ sont supprimés. Cela sert deux objectifs : d'une part, cela limite le risque de dévoiler un document original dans le cas où les données $\mathcal{T}_{c_i}$ sont confidentielles, et d'autre part, cela limite les doublons néfastes à l'apprentissage d'un classifieur dans le cas où les données $\mathcal{G}_{c_i}$ sont utilisées en complément de $\mathcal{T}_{c_i}$. En pratique, cela concerne environ 10 % des textes générés dans nos expériences. Notons que dans le scénario où les données sont confidentielles, la mise à disposition du générateur lui-même n'est pas envisageable (Carlini *et al.*, 2020). Dans les expériences rapportées ci-dessous, ce sont 16 000 textes qui sont ainsi générés pour chaque classe c_i (ce nombre de textes a été fixé arbitrairement).

[1] https://github.com/openai/gpt-2

3.3 A propos de confidentialité

Dans le scénario où les données d'origine ne peuvent pas être distribuées notamment pour des questions de confidentialité, il convient de se demander si des informations sensibles peuvent être retrouvées avec l'approche proposée. Si tout le modèle génératif est mis à disposition, ce risque a été étudié (Carlini *et al.*, 2020), et existe, du moins d'un point de vue théorique dans des conditions particulières[2].

Quand seules les données générées sont mises à disposition, il y a également des risques d'y retrouver des informations confidentielles. Sans autre garde-fou, il est en effet possible que parmi les textes générés, certains soient des paraphrases de morceaux du corpus d'entraînement. Cependant, le risque est très limité :

- tout d'abord, parce qu'il n'y a pas moyen, pour l'utilisateur, de distinguer ces paraphrases parmi toutes les phrases générées ;
- d'autre part, parce qu'en pratique, des mesures supplémentaires peuvent être prises en amont (par exemple, dé-identification du corpus d'entraînement) et en aval (suppression des phrases générées contenant des informations spécifiques ou nominatives...) ;
- enfin, des systèmes plus complexes pour supprimer des paraphrases, tels ceux développés pour les tâches *Semantic Textual Similarity* (Jiang *et al.*, 2020, par exemple), peuvent même être envisagés.

Ces mesures rendent hautement improbable la possibilité d'extraire une information réellement exploitable des données générées.

4 Tâches et jeux de données

Les expériences rapportées dans la section suivante reposent sur deux jeux de données utilisés pour des tâches de classification. L'un est composé de tweets en anglais, l'autre de textes en français. Nous les présentons ci-dessous.

4.1 Classification de textes anglais avec les données MediaEval 2020

Ce jeu de données a été développé pour la détection de fausses informations au sein des réseaux sociaux dans le cadre du challenge FakeNews de MediaEval 2020 (Pogorelov *et al.*, 2020). Dans cette tâche, des tweets sur la 5G ou le coronavirus ont été manuellement annotés selon trois classes $c_i, i \in \{'5G', 'other', 'non'\}$ (Schroeder *et al.*, 2019). $'5G'$ contient les tweets propageant des théories complotistes associant 5G et coronavirus, $'other'$ des tweets propageant d'autres théories complotistes (pouvant porter sur la 5G ou le covid mais sans les associer), et $'non'$ des tweets ne propageant pas de théories complotistes. Il est important de noter que les classes sont déséquilibrées ; ainsi dans le jeu d'entraînement $\mathcal{T}$: $|\mathcal{T}_{5G}| = 1\,076, |\mathcal{T}_{\text{other}}| = 620, |\mathcal{T}_{\text{non}}| = 4\,173$.

[2] Voir également la discussion sur le blog de Google AI.

```
- If the FBI ever has evidence that a virus or some other problem caused or
  contributed to the unprecedented 5G roll out in major metro areas, they need
  to release it to the public so we can see how much of a charade it is when you
  try to downplay the link.
- So let's think about this from the Start. Is it really true that 5G has been
  activated in Wuhan during Ramadan? Is this a cover up for the fact that this is
  the actual trigger for the coronavirus virus? Was there a link between 5G and the
  coronavirus in the first place? Hard to say.
- We don't know if it's the 5G or the O2 masks that are killing people. It's the
  COVID19 5G towers that are killing people. And it's the Chinese people that are
  being controlled by the NWO
```

FIGURE 1 : Exemples de tweets générés artificiellement avec le modèle GPT-2 entraîné sur les données MediaEval avec la classe $\mathcal{T}_{5G}$.

L'augmentation de données est effectuée comme décrit ci-dessus. La figure 1 présente trois exemples de textes générés à partir du jeu de données MediaEval 2020 pour la classe '5G'.

4.2 Classification de textes français avec les données de FLUE

Le deuxième jeu de données que nous utilisons est tiré de la suite d'évaluation pour le français FLUE (Le *et al.*, 2020). Il s'agit de la partie française des données Cross Lingual Sentiment (CLS-FR) (Prettenhofer & Stein, 2010), qui consiste en des commentaires de produits (livres, DVD, musique) sur Amazon. La tâche est de prédire si le commentaire est positif (noté plus de 3 étoiles sur le site marchand) ou négatif (moins de 3 étoiles). Le jeu de données est divisé en ensembles d'entraînement et de test, équilibrés. Dans nos expériences, nous ne distinguons pas les produits : nous n'avons que deux classes (positif, négatif) avec des textes traitant de livres, de DVD ou de musique.

Comme pour les données MediaEval, un modèle de langue est appris pour chacune des nos deux classes à partir des données d'entraînement. La génération est ensuite faite comme décrit dans la section précédente. Des exemples pour la classe commentaires négatifs de CLS-FR sont donnés dans la figure 2.

Comme on peut le voir sur ces exemples (y compris ceux de MediaEval en figure 1), les textes générés semblent respecter la classe attendue (voir section 5.2 pour une discussion de ce point). Pour autant, ils comportent souvent des défauts qui rendent détectable le fait qu'ils aient été générés. C'est plus particulièrement le cas pour les textes en français, ce qui peut s'expliquer par le fait que l'on ne disposait pas, au moment des expériences, de modèle pré-entraîné pour le français ; le modèle, ainsi que le tokeniseur, sont donc basés sur le modèle GPT anglais. Des modèles GPT-2 pour le français sortis très récemment[3] pourraient améliorer cet aspect là.

[3]Par exemple, le modèle Pagnol de LightOn.

```
- Déçue... J'ai eu je l'avoue du mal à lire ce livre arrivé au milieu de celui-ci.
  L'histoire ne paraît pas vraiment très réaliste. Le policier est plus guidé
  par de la chance que par son instinct. Que se serait-il passer s'il n'avait pas
  rencontré cette dame insolite ? Non ! Je ne crois pas que je lirais d'autres
  livres de fred vargas... Dommage je n'ai pas encore trouvé une source infaillible
  de bons polars.
- De la daube. Cet homme ferait mieux de mettre son piano à la benne. Il n'y
  a pas de musicalité, ce disque irrite et agresse, ou au mieux il agresse et
  abuse son timbre et pénible accent amoureux. Musicalement, c'est de la musique
  de... chandler, on se dit... "c'mere irons up". Une chose est surement restée
  disponible sur cet album, mais attention aux maisons de disque !
- Gros navet. Décor atrocement kitch, couleurs d'un mauvais goût abominable qui
  rendrait effleuré un ami en le dire... ça marche. Aucun suspense, tout est
  répétitif, les personnages sont inconséquents, ennuyeux. A eviter absolument.
```

FIGURE 2 : Exemples de commentaires générés artificiellement avec le modèle GPT-2 entraîné sur les données CLS-FR avec la classe $\mathcal{T}_{negatif}$.

5 Expérimentations : approches neuronales

Dans les expériences rapportées ci-dessous, les performances sont mesurées en termes de micro-F1 (équivalent au taux de bonnes classifications), et, pour prendre en compte le déséquilibre des classes (notamment dans le jeu de données MediaEval), en termes de macro-F1 et de MCC (Matthews Correlation Coefficient[4]), tels qu'implémentés dans la bibliothèque scikit-learn. Ces performances sont mesurées sur les jeux de test officiels des tâches MediaEval (Pogorelov *et al.*, 2020) et CLS-FR (Le *et al.*, 2020), bien sûr disjoints des ensembles d'entraînement $\mathcal{T}$.

5.1 Premiers résultats

Pour nos premières expériences, nous utilisons des modèles neuronaux de classification état-de-l'art. Pour les données MediaEval, en anglais, nous optons pour RoBERTa (Liu *et al.*, 2019) pré-entraîné pour l'anglais (modèle *large* avec une couche de classification). C'est ce type de modèle à base de *transformers* qui a obtenu les meilleurs résultats sur ces données lors du challenge MediaEval 2020 (Cheema *et al.*, 2020; Claveau, 2020a). Parmi les variantes de BERT (Devlin *et al.*, 2019), RoBERTa a été ici préféré pour son tokeniseur plus adapté aux spécificités d'écriture très libre que l'on trouve dans les tweets (mélange de majuscules et minuscules, absence ou multiplication de ponctuations, abréviations...). Pour les données CLS-FR de FLUE, nous utilisons FlauBERT dans son modèle *large-cased* (Le *et al.*, 2020). Cela nous permet de nous comparer aux résultats publiés initialement sur ces données.

Nous mesurons les performances selon les divers scénarios d'entraînement : sur les données d'origine $\mathcal{T}$ (ce qui constitue notre *baseline*), sur les données artificielles $\mathcal{G}$, sur les données artificielles et originales. Dans ce dernier cas, nous testons deux stratégies d'entraînement :

- la première, $\mathcal{T} + \mathcal{G}$, mélange les exemples originaux et artificiels,

[4]Également appelé coefficient Φ ; voir la page Wikipedia dédiée.

modèle	MediaEval micro-F1	MediaEval macro-F1	MediaEval MCC	CLS-FR micro-F1	CLS-FR macro-F1	CLS-FR MCC
BERT* / $\mathcal{T}$	79,57	62,66	55,71	95,44	95,42	90,86
BERT* / $\mathcal{G}$	62,68	54,03	39,27	95,13	95,12	90,25
BERT* / $\mathcal{T} + \mathcal{G}$	75,01	58,81	46,37	95,43	95,42	90,89
BERT* / $\mathcal{G}$ puis $\mathcal{T}$	79,89	60,64	52,02	95,76	95,75	91,51

TABLE 1 : Performances (%) de l'approche neuronale sur les données MediaEval et CLS-FR selon les scénarios d'usage des données artificielles (sans filtrage) (cf. sec. 5.1). Les modèles BERT* utilisés sont respectivement RoBERTA et FlauBERT.

- la deuxième, $\mathcal{G}$ puis $\mathcal{T}$, entraîne sur les données artificielles sur les premières *epochs*, puis sur les données originales pour la dernière *epoch*. Cela implémente une sorte de *fine-tuning* sur les données originales après un premier entraînement sur les données artificielles.

L'implémentation que nous utilisons est celle d'HuggingFace (Wolf *et al.*, 2020) avec une taille du batch fixée à 16 et le nombre d'*epochs* fixé à 3 dans tous les scénarios (nombre d'*epochs* optimal pour la *baseline*), sauf le dernier (3 sur $\mathcal{G}$ puis 1 sur $\mathcal{T}$).

Les résultats pour les jeux de données MediaEval et CLS-FR sont reportés dans le tableau 1. Sur les donnés CLS-FR, on observe très peu de différences entre les différents scenarii et par rapport à la baseline (et notre baseline est tout à fait en ligne avec les résultats état-de-l'art (Le *et al.*, 2020)). La tâche de classification, relativement simple, permet visiblement de générer des données d'aussi bonne qualité que les données originales, menant à des résultats comparables. Sur ce type de tâche, l'utilisation de données générées artificiellement peut donc se faire sans perte de performances.

Les données MediaEval sont plus difficiles comme on peut le voir avec les résultats de la baseline (RoBERTA / $\mathcal{T}$). Sur ces données, dans un scénario de substitution (i.e. quand les données générées servent seules de données d'entraînement), les résultats sont fortement dégradés par rapport à un système entraîné sur les données originales. Cela s'explique bien sûr par le fait que les données générées par chacun des modèles de langue peuvent ne pas appartenir à la classe attendue, les modèles ne capturant pas complètement la spécificité des données de *fine-tuning*. Dans un scénario de complément des données d'apprentissage, l'impact est moins important, particulièrement si les données artificielles sont utilisées uniquement sur les premières *epochs*.

5.2 Résultats avec filtrage automatique

Comme nous l'avons vu, les exemples $\mathcal{G}$ générés par nos modèles GPT-2 peuvent contenir des textes ne relevant pas des classes attendues. Filtrer ou annoter manuellement ces textes est bien sûr possible mais reste une tâche coûteuse. Pour diminuer l'effet de ces textes sur la classification à moindre coût, nous proposons de les exclure à l'aide d'un premier classifieur appris sur les données originales $\mathcal{T}$: tout texte de $\mathcal{G}_{c_i}$ qui n'est pas classé c_i par le classifieur est exclu. On espère ainsi éliminer, automatiquement, les cas les plus évidents de texte artificiels problématiques. Dans les expériences suivantes, nous utilisons le classifieur RoBERTa entraîné sur $\mathcal{T}$ (évalué en première ligne de tab. 1). Ce sont ainsi 40 % des exemples qui sont supprimés. Les exemples artificiels gardés sont notés $\mathcal{G}^f$.

Les résultats avec ces nouveaux jeux épurés d'exemples artificiels dans les mêmes scénarios d'en-

modèle	MediaEval micro-F1	MediaEval macro-F1	MediaEval MCC	CLS-FR micro-F1	CLS-FR macro-F1	CLS-FR MCC
BERT* / $\mathcal{T}$	79,57	62,66	55,71	95,44	95,42	90,86
BERT* / $\mathcal{G}^f$	76,22	64,18	52,75	95,76	95,75	91,51
BERT* / $\mathcal{T} + \mathcal{G}^f$	80,12	66,08	57,44	95,99	95,98	91,97
BERT* / $\mathcal{G}^f$ puis $\mathcal{T}$	83,55	67,90	60,05	95,96	95,95	91,96

TABLE 2 : Performances (%) de l'approche neuronale sur les données MediaEval et CLS-FR selon les scénarios d'usage des données artificielles après filtrage (cf. sec. 5.2). Les modèles BERT* utilisés sont respectivement RoBERTA et FlauBERT.

traînement sont présentés dans le tableau 2 pour les données MediaEval et CLS-FR. On constate que cette stratégie de filtrage se révèle payante, les performances étant améliorées sur l'ensemble des métriques par rapport à l'absence de filtrage. Dans le scénario de substitution, les performances s'approchent désormais de la *baseline*, et sont même meilleures sur la macro-F1 ; cela s'explique par le fait que le jeu artificiel $\mathcal{G}$ est bien plus équilibré que $\mathcal{T}$ et donc plus performant sur les classes minoritaires du jeu de test. Dans le scénario de complément, on observe une amélioration significative par rapport à la *baseline*, notamment avec la stratégie séquentielle.

5.3 Différences entre les classifieurs

Au-delà des mesures de performances globales, il peut être intéressant de vérifier si le classifieur entraîné sur les données artificielles permet de prendre les mêmes décisions qu'un classifieur entraîné sur $\mathcal{T}$. Pour ce faire, on peut regarder la proportion d'exemples (du jeu de test) pour lesquels la décision entre BERT* / $\mathcal{T}$ et BERT* / $\mathcal{G}^f$ diffère. Pour les données CLS-FR, les classifieurs s'accordent sur une grande majorité d'exemples. La figure 3 présente la matrice de confusion des classifieurs FlauBERT / $\mathcal{T}$ et FlauBERT / $\mathcal{G}^f$ sur les données CLS-FR.

À partir de cette matrice de confusion, on peut remarquer que les classifieurs s'accordent effectivement sur la majorité des exemples. Les cas de désaccords sont proportionnellement plus importants sur les faux positifs et faux négatifs, mais même pour ces catégories, on constate tout de même beaucoup d'erreurs communes (resp. 42 et 77 exemples pour les faux positifs et faux négatifs). Les classifieurs ont donc non seulement des performances comparables, mais des comportements très similaires dans le détail puisqu'ils donnent la même classe sur la plupart des exemples.

6 Expérimentations : approches sac-de-mots

Nous testons également des classifieurs reposant sur des représentations sac-de-mots ; nous ne présentons que les résultats de la régression logistique (LR) qui a donné les meilleurs résultats. En général moins performants que les approches à base de *transformers*, ces classifieurs permettent cependant une meilleure explicabilité (Miller, 2018; Carvalho *et al.*, 2019, pour une définition et une caractérisation des méthodes d'apprentissage), par exemple en examinant les poids de régression associés aux mots. Ils sont aussi moins coûteux à entraîner.

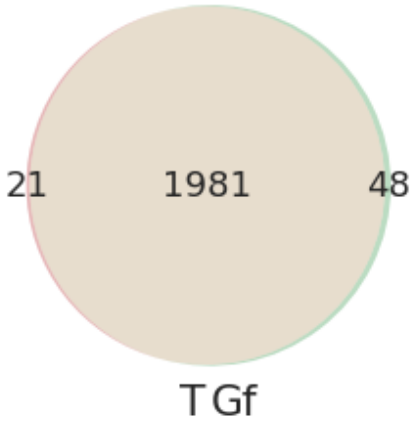
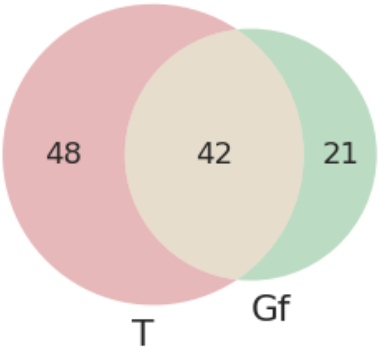
	Positif (vérité-terrain)	Négatif (vérité terrain)
Prédit positif		
Prédit négatif		

TABLE 3 : Matrice de confusion des modèles FlauBERT / $\mathcal{T}$ et FlauBERT / $\mathcal{G}^f$ sur les données CLS-FR. Les diagrammes de Venn font apparaître les proportions d'exemples en commun pour chacune des catégories.

6.1 Premiers résultats

L'implémentation utilisée est celle de scikit-learn (Pedregosa *et al.*, 2011), les textes sont vectorisés avec la pondération TF-IDF et normalisés L2, et les paramètres de la LR sont ceux par défaut sauf pour les suivants : stratégie multiclasse *one-vs.-rest*, nombre d'itérations = 2500. Les résultats des mêmes scénarios que précédemment sont présentés pour les tâches MediaEval et CLS-FR dans les tableaux 4 et 5.

Pour ce type de classifieur, l'intérêt des données générées apparaît pour les deux scénarios et sur nos deux jeux de données. Dans le cas de la substitution, les classifieurs sont légèrement meilleurs que ceux entraînés sur les données originales. Cela démontre l'intérêt de disposer d'une plus grande quantité de données permettant de capturer des variantes de forme dans les textes (synonymes, paraphrases...) que les représentations sac-de-mots ne peuvent sinon pas capturer aussi facilement que les représentations par plongements (pré-entraînées). Dans le scénario où les données sont utilisées en complément, l'augmentation de performances est encore plus marquée et s'approche ainsi de la *baseline* neuronale, tout en ayant les avantages d'un classifieur jugé plus interprétable.

modèle	micro-F1	macro-F1	MCC
LR / $\mathcal{T}$	72,68	56,35	42,22
LR / $\mathcal{G}^f$	74,00	59,18	44,39
LR / $\mathcal{T} + \mathcal{G}^f$	75,46	59,64	45,83

TABLE 4 : Performances (%) de l'approche LR/sac-de-mots sur les données MediaEval selon les scénarios d'usage des données artificielles filtrées : sans, par substitution, en complément.

modèle	micro-F1	macro-F1	MCC
LR / $\mathcal{T}$	84,77	84,70	69,48
LR / $\mathcal{G}^f$	87,16	87,14	74,27
LR / $\mathcal{T} + \mathcal{G}^f$	88,36	88,34	76,69

TABLE 5 : Performances (%) de l'approche LR/sac-de-mots sur les données CLS-FR selon les scénarios d'usage des données artificielles filtrées : sans, par substitution, en complément.

6.2 Effet de la qualité des donnés générées

On peut se demander quelle est l'influence de la qualité des données générées (mêmes filtrées) sur les résultats du classifieur final (cf. section 5.2). Pour étudier cela, nous injectons du bruit dans la classification pour simuler des filtrages faits avec des classifieurs de qualité variable. Cela est fait simplement en remplaçant, pour des exemples de $\mathcal{G}^f$ tirés au hasard, la classe prédite (par le générateur et par le classifieur filtrant) par une classe tirée aléatoirement. Le nombre d'exemples subissant ce traitement est calculé pour que la probabilité d'erreurs ainsi ajoutées fasse chuter le taux de bonnes de précision à 80 %, 70 %, etc. L'effet de ces filtrages sur les performances finales des stratégies complément et substitution sont présentés dans la figure 3 (données MediaEval) avec la régression logistique comme classifieur final.

Comme on peut le constater dans cette figure, ces résultats empiriques sur l'influence de la qualité du filtrage sont sans surprise. Dans le scénario substitution, la performance finale est fortement dépendante de la qualité du classifieur filtrant ; dans le cas présent, on atteint des performances équivalentes au jeu de données originales quand le taux de bonnes classifications du filtre dépasse 70 %. Dans le cas du scénario complément, le gain est sensible dès que le filtre a un taux de bonnes classifications supérieur au hasard.

Conclusion et perspectives

Dans un scénario où les données originales ne peuvent pas être distribuées, nous avons montré qu'il était possible de générer des données artificielles à des fins d'apprentissage supervisé. Pour les classifieurs état-de-l'art à base de *transformers*, cela dégrade les performances (par rapport à celles atteintes avec les données originales) mais dans une proportion contenue (-4 % de taux de bonnes classifications). En revanche, pour les classifieurs exploitant des représentations sac-de-mots, on constate une amélioration portée par la plus grande quantité de données d'apprentissage disponibles.

Dans un scénario où les données artificielles viennent en complément des données originales, nous avons montré que les classifieurs bénéficiaient de l'apport de données supplémentaires, y compris les

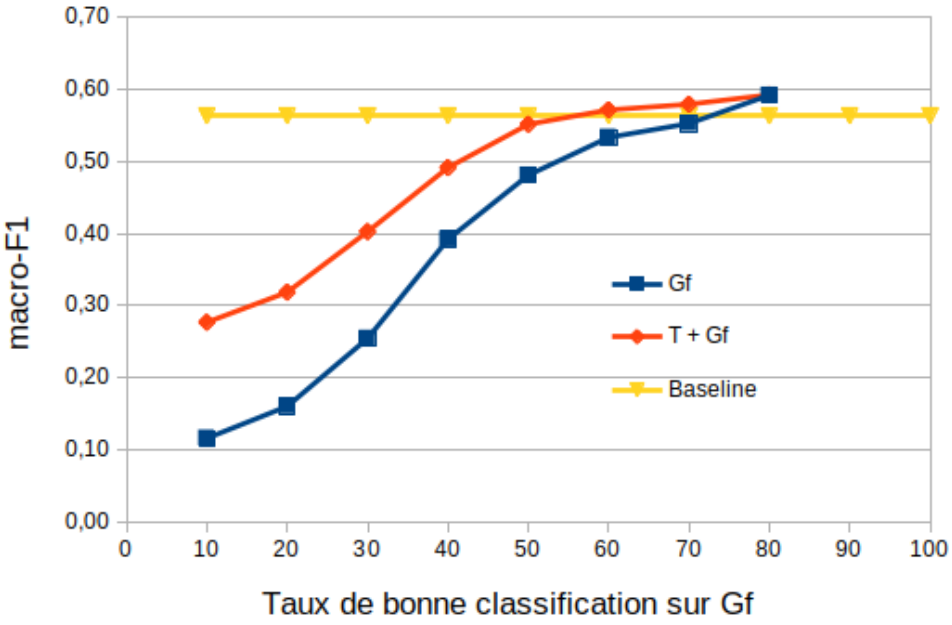


FIGURE 3 : Performances (macro-F1) en fonction de la qualité (taux de bonnes précisions en %) du classifieur servant au filtrage des données générées automatiquement ; données MediaEval avec la classifieur régression logistique.

réseaux de neurones. Ce résultat est particulièrement positif pour les approches sac-de-mots, plus sensibles aux reformulations, qui bénéficient clairement de l'ajout de ces exemples artificiels. On a ainsi un bon compromis entre méthodes rapides à entraîner, plus facilement interprétables, tout en ayant des performances proches des réseaux de neurones.

Comme nous l'avons vu, ces résultats sont obtenus à condition que les données générées soient filtrées au préalable, ce qui semble contredire plusieurs travaux cités en sec. 2. Dans nos expériences, elles l'ont été automatiquement ; une correction manuelle des données (de leurs classes) est aussi envisageable et permettra de meilleurs résultats, mais avec un coût d'annotation supplémentaire. L'emploi de ces méthodes à d'autres données et d'autres tâches de TAL que la classification de texte reste une piste prometteuse. Parmi ces tâches de TAL, celles reposant sur de l'étiquetage de mots posent des problèmes différents et nécessitent des solutions adaptées. Dans le futur, il serait intéressant de vérifier la consistance de nos résultats selon d'autres approches de génération (Kumar *et al.*, 2020). Il semble également intéressant d'étudier plus profondément l'impact de la qualité du classifieur servant à filtrer les données artificielles. De plus, l'intégration de cette étape de filtrage comme une contrainte lors de la génération des exemples artificiels est une piste prometteuse.

À des fins de réplicabilité, les scénarios d'entraînement présentées dans cet article sont accessibles en ligne pour les données MediaEval et CLS-FR. La génération des exemples repose sur `https://github.com/minimaxir/gpt-2-simple`. Les données sont accessibles auprès de leurs producteurs (voir section 4).

Références

AMIN-NEJAD A., IVE J. & VELUPILLAI S. (2020). Exploring transformer text generation for

medical dataset augmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, p. 4699–4708, Marseille, France : European Language Resources Association.

CARLINI N., TRAMER F., WALLACE E., JAGIELSKI M., HERBERT-VOSS A., LEE K., ROBERTS A., BROWN T., SONG D., ERLINGSSON U., OPREA A. & RAFFEL C. (2020). Extracting training data from large language models. *arXiv*.

CARVALHO D. V., PEREIRA E. M. & CARDOSO J. S. (2019). Machine learning interpretability : A survey on methods and metrics. *Electronics*, **8**(8). DOI : 10.3390/electronics8080832.

CHEEMA G. S., HAKIMOV S. & EWERTH R. (2020). TIB's Visual Analytics Group at MediaEval '20 : Detecting Fake News on Corona Virus and 5G Conspiracy. In *MediaEval Benchmarking Initiative for Multimedia Evaluation (MediaEval 2020)*, online, United States.

CLAVEAU V. (2020a). Detecting fake news in tweets from text and propagation graph : IRISA's participation to the FakeNews task at MediaEval 2020. In *MediaEval Benchmarking Initiative for Multimedia Evaluation (MediaEval 2020)*, online, United States. HAL : hal-03116027.

CLAVEAU V. (2020b). Query expansion with artificially generated texts. *CoRR*, **abs/2012.08787**.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

JIANG H., HE P., CHEN W., LIU X., GAO J. & ZHAO T. (2020). SMART : Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 2177–2190, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.197.

JUNGIEWICZ M. & SMYWINSKI-POHL A. (2019). Towards textual data augmentation for neural networks : synonyms and maximum loss. *Computer Science*, **20**(1). DOI : 10.7494/csci.2019.20.1.3023.

KOBAYASHI S. (2018). Contextual augmentation : Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 452–457, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-2072.

KUMAR V., CHOUDHARY A. & CHO E. (2020). Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, p. 18–26, Suzhou, China : Association for Computational Linguistics.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBE B., BESACIER L. & SCHWAB D. (2020). FlauBERT : Unsupervised Language Model Pre-training for French. In *LREC*, Marseille, France. HAL : hal-02890258.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach.

MILLER T. (2018). Explanation in artificial intelligence : Insights from the social sciences. *Artificial Intelligence*, **267**.

MUELLER J. & THYAGARAJAN A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2786–2792 : AAAI Press.

PAPANIKOLAOU Y. & PIERLEONI A. (2020). Dare : Data augmented relation extraction with gpt-2.

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.

POGORELOV K., SCHROEDER D. T., BURCHARD L., MOE J., BRENNER S., FILKUKOVA P. & LANGGUTH J. (2020). Fakenews : Corona virus and 5g conspiracy task at mediaeval 2020. In *MediaEval 2020 Workshop*.

PRETTENHOFER P. & STEIN B. (2010). Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 1118–1127, Uppsala, Sweden : Association for Computational Linguistics.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*.

SCHROEDER D. T., POGORELOV K. & LANGGUTH J. (2019). Fact : a framework for analysis and capture of twitter graphs. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, p. 134–141 : IEEE.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 30*, p. 5998–6008. Curran Associates, Inc.

WEI J. & ZOU K. (2019). EDA : Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 6382–6388, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1670.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 38–45, Online : Association for Computational Linguistics.

WU X., LV S., ZANG L., HAN J. & HU S. (2019). Conditional bert contextual augmentation. In J. M. F. RODRIGUES, P. J. S. CARDOSO, J. MONTEIRO, R. LAM, V. V. KRZHIZHANOVSKAYA, M. H. LEES, J. J. DONGARRA & P. M. SLOOT, Éds., *Computational Science – ICCS 2019*, p. 84–95, Cham : Springer International Publishing.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Open Information Extraction: Approche Supervisée et Syntaxique pour le Français

Massinissa Atmani[1, 2] Mathieu Lafourcade[1]
(1) LIRMM, 860 rue de St Priest, 34095 Montpellier, France
(2) Amaris Research Unit, 25 boulevard Eugène Deruelle, 69003 Lyon, France
massinissa.atmani@etu.umontpellier.fr, mathieu.lafourcade@lirmm.fr

RÉSUMÉ

L' Open Information Extraction, est un paradigme d'extraction conçu pour gérer l'adaptation de domaine, la principale difficulté des approches traditionnelles pour l'extraction d'informations. Cependant, la plupart des approches se concentrent sur l'anglais. Ainsi, nous proposons une approche supervisée pour l'OpenIE pour le français, nous développons également un corpus d'entraînement et un référentiel d'évaluation. Nous proposons un nouveau modèle basé en deux étapes pour l'étiquetage de séquence, qui identifie d'abord tous les arguments de la relation avant de les étiqueter. Les expérimentations montrent non seulement que l'approche que nous proposons obtient les meilleurs résultats, mais aussi que l'état de l'art actuel n'est pas assez robuste pour s'adapter à un domaine différent du domaine du corpus d'entraînement.

ABSTRACT

Supervised Syntactic Approach for French Open Information Extraction.

Most of Open Information Extraction approaches focus on English. Hence, we propose a supervised OpenIE for French, we also derive a training set and an evaluation benchmark for French OpenIE. We propose a new two-stage pipeline model for sequence labeling, that first identifies all the arguments of the relation and only then classifies them according to their most likely label. The experiments not only show that our proposed approach achieves the best results, but also that the current state-of-the-art approach is not cross-domain friendly and fails when facing out-of-domain data (their domain is different from the training-set's domain).

MOTS-CLÉS : Extraction d'information, Apprentissage machine, Syntaxe.

KEYWORDS: Information Extraction, Machine Learning, Syntax.

1 Introduction

L'Open Information Extraction (OpenIE) (Yates *et al.*, 2007) consiste à extraire des faits et des événements exprimés dans une phrase, à travers une représentation prédicat-argument. (Yates *et al.*, 2007) le présente comme *"un nouveau paradigme d'extraction qui facilite l'extraction de relations à partir de texte en considérant* ***l'indépendance de domaine*** *et qui s'adapte facilement à la diversité et à la taille du corpus du Web"*. De nombreuses tâches de TALN (Mausam, 2016) ont bénéficié de l'OpenIE telles que les réponses aux questions avec documents multiples (Fan *et al.*, 2019), l'induction de schéma d'événement (Balasubramanian *et al.*, 2013) et la génération de vecteur de mots (Stanovsky *et al.*, 2015).

Compte tenu de l'inexistence d'approches qui se focalisent sur l'OpenIE pour le Français, nous proposons une approche supervisée pour le Français à base de réseaux neuronaux. Pour ce faire, nous construisons un corpus d'entraînement pour le Français en traduisant des corpus Anglais exploités par (Corro & Gemulla, 2013). Nous annotons également un référentiel d'évaluation issu d'articles de journaux du domaine de la finance, qui est différent du domaine du corpus d'entraînement pour vérifier le critère d'indépendance de domaine.

Notre modèle proposé consiste en deux sous-modules faiblement couplés, le premier module est un modèle multi-tâches qui extrait la relation de prédicat, puis cherche à trouver tous les arguments étant donné la relation de prédicat extraite. Le deuxième module prend en entrée le prédicat et les arguments extraits, puis attribue l'étiquette ou tag le plus probable à chaque argument identifié tel que sujet, objet, argument temporel ou argument de localisation. La raison d'une telle approche découle des tendances récentes dans l'analyse des dépendances syntaxiques neuronales (Dozat & Manning, 2016), qui consiste à trouver la structure de dépendance syntaxique non étiquetée (topologie de l'arbre syntaxique), et seulement ensuite attribuer une étiquette pour chaque arc prédit de l'arbre. Plus précisément, pour chaque paire de mots leur modèle calcule la probabilité d'existence d'un arc reliant ces deux mots ainsi qu'une étiquette de fonction syntaxique pour chaque arc de l'arbre syntaxique. Contrairement à leur approche, nous ne calculons que la probabilité entre un mot et la plage de mots représentant la phrase du prédicat extraite à l'étape précédente. Dans notre configuration, les arcs prédits indiquent les arguments extraits de la relation de prédicat, ces arguments extraits seront raffinés et étiquetés à l'étape suivante.

Finalement, les résultats des expérimentations montrent que les approches basées sur le modèle de langage BERT (Devlin *et al.*, 2019) sont beaucoup moins performantes sur des échantillons de données issues de domaines qui sont différents de celui sur lequel le modèle de langage a été entraîné.

2 État de l'art d'OpenIE

2.1 Première génération

Les premiers systèmes OpenIE n'exploitaient qu'une analyse syntaxique basique telle que l'étiquetage grammatical et l'extraction terminologique (*chunking*) (Yates *et al.*, 2007; Fader *et al.*, 2011). Des systèmes plus avancés ont considérablement amélioré les performances en exploitant un traitement linguistique plus avancé. (Corro & Gemulla, 2013) ont utilisés l'arbre d'analyse syntaxique des dépendances pour décomposer des phrases complexes en un ensemble de clauses indépendantes, où chaque type de clause peut exprimer une extraction avec une structure prédicat-arguments prédéfinis. Le *Semantic Role Labelling* (SRL) consiste à étiqueter les mots d'une phrase avec leur rôle sémantique, tel que agent, thème et artefact. La tâche SRL est quelque peu similaire à la tâche OpenIE, et en raison de la disponibilité des ressources, (Christensen *et al.*, 2010) ont utilisés un analyseur SRL pour dériver leur système *SRLIE*.
Plusieurs systèmes d'OpenIE extraient des relations exprimées par des verbes et ignorent les relations nominales. (Yahya *et al.*, 2014) ont proposés *RENOUN* pour extraire les relations nominales. (Pal & Mausam, 2016) ont conçus un système OpenIE adapté aux relations exprimées par des démonymes et des noms composés relationnels.
OPENIE4 a été conçu de la fusion des systèmes *SRLIE* (Christensen *et al.*, 2010) et *RelNoun* (Pal & Mausam, 2016). Ils ont augmentés *OpenIE4* avec un système d'OpenIE adapté aux relations numériques ainsi qu'un système analysant les conjonctions de coordination afin de concevoir *OpenIE5*.

(Gotti & Langlais, 2016) ont proposés un système d'OpenIE pour le français en adaptant le système *Reverb* (Fader *et al.*, 2011) afin qu'il extrait des fait simples à partir de Wikipédia français.

2.2 OpenIE multilingue

La plupart des systèmes OpenIE pour les langues autres que l'anglais sont des approches ad-hoc à base de règles, avec des performances assez limitées. Parmi ces approches, deux systèmes se distinguent : ArgOIE et PredPatt. (Gamallo & Garcia, 2015) présentent ArgOIE qui prend comme entrée l'analyse syntaxique de dépendances au format CoNLL-X, identifie les structures d'argument dans l'analyse des dépendances et extrait un ensemble de propositions basique de chaque structure d'argument. ArgOIE supporte l'OpenIE dans les trois langues : anglais, espagnol et portugais. Similaire à ArgOIE, PredPatt (White *et al.*, 2016) prend lui aussi en entrée l'analyse syntaxique de dépendances au format Universal Dependency (Nivre *et al.*, 2016) et retourne un ensemble de structures prédicat-arguments en appliquant des patterns syntaxique et peut en principe supporter toutes les langues supportés par Universal Dependency.
(Ro *et al.*, 2020) ont proposé Multi2OIE, un modèle d'étiquetage de séquence pour OpenIE, qui prédit d'abord tous les prédicats de relation en utilisant BERT, puis prédit les arguments sujet et objet associés à chaque relation en utilisant des blocs d'attention multi-têtes. Plus précisément, ils utilisent la version multilingue de BERT afin de supporter l'OpenIE dans toutes les langues supportées par BERT-Multilingue. Leur approche à l'avantage de pouvoir s'adapter aux différentes langues sans aucune langue pivot, puisque leur modèle est entraîné seulement sur un corpus en anglais.

3 Méthodologie

Nous présentons notre méthode en détail dans cette section. Tout d'abord, nous présentons la manière dont nous formulons la tache de l'OpenIE ainsi qu'un aperçu de notre approche supervisée de l'OpenIE dans 3.1 et 3.2 respectivement. Enfin, nous décrivons la représentation des entrées et notre nouvelle architecture pour notre modèle d'OpenIE respectivement dans 3.3 & 3.4.

3.1 Définition du Problème

Étant donné une phrase $S = (w_1, w_2, ..., w_n)$, nous dérivons d'abord l'arbre syntaxique des dépendances pour obtenir l'étiquetage grammatical et les relations de dépendance syntaxique.
Nous transmettons ces plongements au modèle pour produire une balise de séquence $T = (y_1, y_2, ..., y_n)$, avec l'ensemble de balises $Y = \{A0, P, A1, A2, O\}$. La séquence produite représente le tuple ($A0$: sujet, P : prédicat, $A1$: objet ...) au format modèle BIES (*Begin, Inside, End, Single*).

TABLE 1 – Exemple de la représentation de la sortie du modèle.

Phrase	Brady tente d' appeler le Shérif .
Séquence d'étiquettes	$A0_S\ P_B\ P_I\ P_E\ A1_B\ A1_E\ O$
Représentation de la sortie	Brady$_{A0_S}$ tente$_{P_B}$ d'$_{P_I}$ appeler$_{P_E}$ le$_{A1_B}$ Shérif$_{A1_E}$.$_O$
Relation	($A0$:Brady, P :tente d' appeler, $A1$:le Shérif)

3.2 Approche

S'inspirant de (Stanovsky *et al.*, 2018), nous abordons la tâche d'OpenIE comme un problème d'étiquetage de séquence (Sequence Labelling) avec le format d'étiquetage BIES (Begin, Inside, End, Single). A partir d'une phrase donnée $S = (w_1, w_2, ..., w_n)$, l'étiquetage de séquence vise à assigner à chaque mot de la phrase l'étiquette le plus probable, donnant lieu à une séquence de labels $T = (y_1, y_2, ..., y_n)$. On extrait une relation à la fois, en considérant à chaque itération un mot de la phrase comme potentiel prédicat de la relation, à partir duquel on déduit un masque binaire $M = (m_1, m_2, ..., m_n)$.

3.3 Représentation des entrées

Nos deux sous-modules exploitent les mêmes entrées, à l'exception du modèle d'argument qui attend les bornes inféré par le modèle de prédicat, et utilise un masque différent pour modéliser la topologie de l'arbre syntaxique.
Nous utilisons la bibliothèque Stanza (Qi *et al.*, 2020) pour obtenir l'étiquetage grammatical (POS) et l'arbre d'analyse syntaxique (relations de dépendances), avec la représentation Universal Dependency (Nivre *et al.*, 2016).
Les vecteurs de l'étiquetage grammatical et des relations syntaxiques sont obtenues en utilisant l'encodage *one-hot encoding* (encodage 1 parmi n) où chaque catégorie est assignée à un vecteur différent.

3.4 Extraction de la structure Prédicat-Argument

Notre premier sous-module extrait la représentation prédicat-argument tout en ignorant l'étiquette ou le type des arguments. Par conséquent, le sous-module est optimisé vis-à-vis de deux tâches : l'extraction de relation de prédicat et l'identification des arguments. La dernière tâche dépendant de la sortie de la tâche précédente.
Les entrées pour le sous-module sont la concaténation des trois vecteurs de caractéristiques : $E_{pos}, E_{dep}, E_{masque}$. Le premier représente le plongement vectoriel de l'étiquetage grammatical, le second représente le plongement vectoriel de la relation syntaxique et le troisième représente le masque de prédicat binaire.
Puisque nous extrayons une relation à la fois, E_{masque} est un simple vecteur binaire pour indiquer quel mot de la phrase est le prédicat candidat. Le sous-module partage une couche Bi-LSTM pour les deux tâches et exploite une couche de champs aléatoires conditionnels (CRF) (Lafferty *et al.*, 2001) pour chaque tâche.
Étant donné une instance d'entrée (S, M) avec S une phrase et M un vecteur binaire (0 et 1), pour chaque mot $w_i \in S$ nous calculons un vecteur de caractéristiques :

$$x_i = E_{pos}(w_i) \oplus E_{dep}(w_i) \oplus E_{masque}(w_i) \tag{1}$$

Le vecteur de caractéristiques 1 est transmis au Bi-LSTM, qui calcule une représentation conceptualisée bidirectionnelle de chaque mot de la phrase (le contexte précédent (*forward*) et le contexte suivant (*backward*) de chaque mot).
Après, la moyenne des représentations conceptualisées du contexte précédent et suivant du Bi-LSTM est calculée pour chaque mot et est transmise à une couche dense.

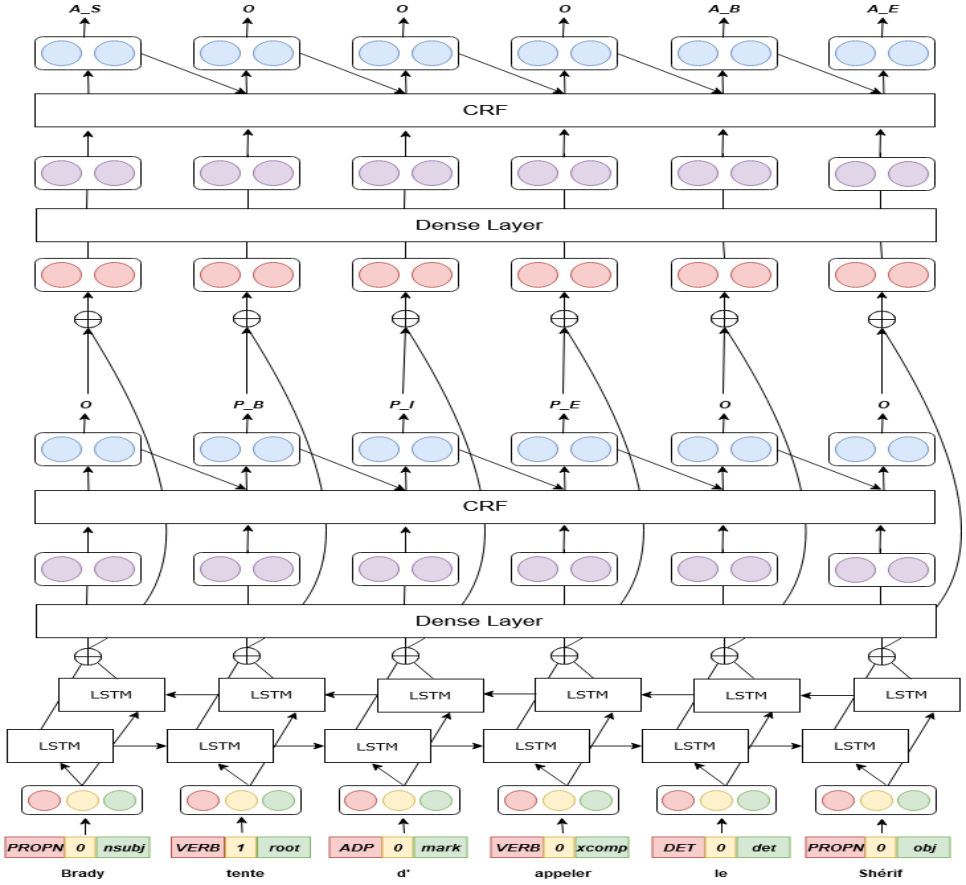
$$v_i^{\rightarrow}, v_i^{\leftarrow} = Bi - LSTM(x_i) \tag{2}$$

$$u_i = AVG(v_i^{\rightarrow}, v_i^{\leftarrow}) \tag{3}$$

$$h_i = Wu_i + b \tag{4}$$

Ensuite, la représentation est transmise au décodeur de chaque tâche. Puisque les deux tâches utilisent comme décodeur les champs aléatoires conditionnels (CRF) , nous introduisons le décodeur CRF.

FIGURE 1 – Extraction des prédicat-arguments



3.4.1 Décodeur CRF

Étant donné la séquence d'entrée du décodeur $H = \{h_i\}_{i=1}^{n}$ et une séquence d'étiquettes $Y = \{y_i\}_{i=1}^{n}$, le décodeur calcule le score de décodage $S(H, Y)$.

$$S(H, Y) = \sum_{i=1}^{n-1} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} H_{i, y_i} \tag{5}$$

H est une matrice d'émission $n \times k$, où n est la longueur de la séquence, k le nombre d'étiquettes distinctes et H_{ij} est le score de j-ème tag à la position i de la séquence. A est une matrice de transition $k \times k$, où A_{ij} représente le score de transition du i-ème tag vers le j-ème tag.
Puis $p(Y|H)$ est calculé, une probabilité conditionnelle sur toutes les séquences d'étiquettes possibles Y en utilisant Softmax, où Y_H représente les séquences d'étiquettes possibles pour H.

$$p(Y|H) = \frac{e^{S(H,Y)}}{\sum_{Y' \in Y_H} e^{S(H,Y')}} \tag{6}$$

Lors du décodage, nous recherchons la séquence ayant le score maximum y^*, en utilisant l'algorithme de Viterbi (Forney, 1973).

$$y^* = argmax_{Y \in Y_H} S(H, Y) \tag{7}$$

La sortie de l'encodeur 4 est d'abord transmise à l'extracteur de prédicat, qui identifie le prédicat. Après avoir extrait le prédicat 7, la phrase de prédicat est transmise à l'extracteur d'arguments en tant que vecteur binaire qui indique les bornes du prédicat extrait. Enfin, la sortie de l'encodeur 4 est concaténée avec la sortie du décodeur CRF du prédicat 7 et est envoyée au décodeur CRF de l'extracteur d'arguments. La nouvelle représentation est donnée par l'équation suivante :

$$h_i(Argument) = h_i \oplus y_i(Predicat) \tag{8}$$

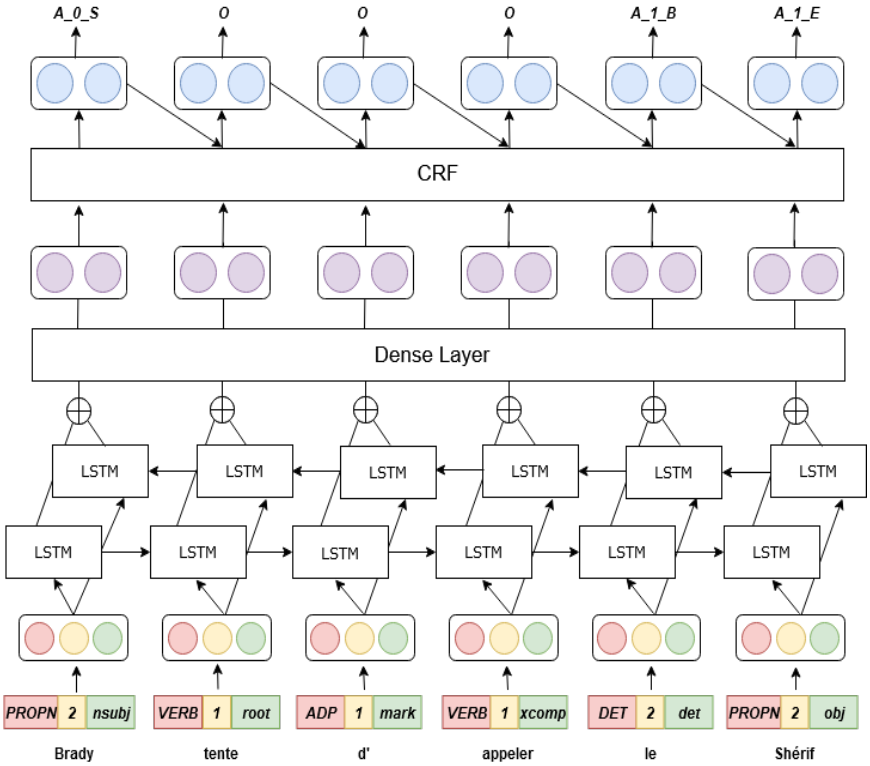
Les deux tâches sont optimisées conjointement et nous maximisons la vraisemblance logarithmique de la séquence d'étiquettes correcte de chaque tâche sur l'ensemble d'apprentissage $\{(H_j, Y_j)\}$, en minimisant la fonction de coût : la Negative Log Likelihood (NLL) (Yao *et al.*, 2019).

$$NLL = -\sum_j \log p(Y|H) \tag{9}$$

La fonction de coût du sous-module est simplement la somme de la fonction de coût de chaque tâche :

$$NLL = -\sum_j \log p(Y|H)_{predicat} - \sum_j \log p(Y|H)_{argument} \tag{10}$$

FIGURE 2 – Étiqueteur des arguments



3.5 Étiquetage des arguments

Après l'extraction de la structure prédicat-argument, le premier sous-module alimente le prédicat et les arguments extraits par le deuxième sous-module. L'entrée du modèle est le vecteur de caractéristiques défini par 11, et consiste en la concaténation des vecteurs représentant le plongement vectoriel de

l'étiquetage grammatical, le plongement vectoriel de la relation de dépendance syntaxique et E_{pr-arg}, le vecteur inféré dans la première étape qui représente le prédicat et les arguments extraits.

$$x_i = E_{pos}(w_i) \oplus E_{dep}(w_i) \oplus E_{pr-arg}(w_i) \tag{11}$$

Le sous-module exploite la même architecture que le premier sous-module qui consiste en un décodeur CRF empilé sur une couche Bi-LSTM et cherche à attribuer l'étiquette la plus probable aux arguments extraits lors de la première étape. Comme l'extracteur d'arguments de prédicat, le modèle est optimisé pendant l'entraînement en minimisant la Negative Log Likelihood.

4 Expérimentations

Dans cette section, les corpus de données d'entraînement et les hyper-paramètres sont respectivement présentés dans 4.2 et 4.1, puis 4.3 et 4.4 décrivent le référentiel d'évaluation et la stratégie d'évaluation. Finalement, nous présentons l'étude d'impact de l'architecture et les systèmes de référence dans 4.5 et 4.6.

4.1 Hyperparameters

Le tableau 2 ci-dessous, reprend les hyper-paramètres de notre modèle, qui sont les mêmes pour les deux sous-modules. Nous avons entraîné notre modèle à l'aide de l'optimiseur Adam.

TABLE 2 – hyper-paramètres

hyper-paramètres du modèle	
Taille de l'état caché du LSTM	128
Dropout de l'état récurrent du LSTM	0.3
Dropout de l'entrée du LSTM	0.3
Dropout de la sortie du LSTM	0.3
Dropout du vecteur	0.1
Regularization L2	0.001
Taille du vecteur	20
Taille du batch	5
Taux d'apprentissage	0.001
Nombre des Hyper-Paramètres	
Extraction de la structure Prédicat-Argument	590,553
Étiquetage des arguments	592,911
Model complet	1,183,464

Dataset	Domaine	#Phrases	#Relations
ReVerb	Yahoo	500	1 551
NYT	New York Times	200	642
Wiki	Wikipedia	200	568

TABLE 3 – Données d'apprentissage

4.2 Données d'apprentissage

Comme aucun corpus n'existe pour le Français dans la tâche d'OpenIE, nous décidons de construire nous même un corpus d'apprentissage pour le Français. Pour ce faire, nous optons pour une approche

semi-supervisée dans laquelle nous traduisons automatiquement des corpus en anglais (Corro & Gemulla, 2013) vers le Français en utilisant l'API de Google [1], nous faisons une deuxième passe pour manuellement corriger les erreurs de traduction. La description du corpus obtenu est décrite dans le tableau 3.

4.3 Référentiel d'évaluation

Au vu de l'absence de référentiel d'évaluation pour le Français, nous annotons aussi un référentiel d'évaluation en prenant des phrases issues d'articles de journaux du domaine de la finance, et qui ont été décrit dans (Jabbari *et al.*, 2020). Nous avons décidé de choisir un domaine différent de celui des données d'apprentissage qui couvre principalement des phrases issues du Web. Pour annoter le corpus, nous suivons les recommandations d'annotation de (Lechelle *et al.*, 2019), qui ont aussi été suivis par (Bhardwaj *et al.*, 2019) pour construire CARB : le référentiel d'évaluation pour l'anglais. Cependant, nous avons observé quelques annotations et extractions complexes ou ambiguës :

— Conjonction de coordination : (Lechelle *et al.*, 2019) préconisent de séparer les conjonctions de coordination dans les arguments pour générer plusieurs extractions, sauf que la conjonction de coordination avec le connecteur *et* peut faire l'objet de deux interprétations, l'une cumulative (parfois dite collective) et l'autre distributive. Dans le cas d'une conjonction de coordination cumulative, nous avons trouvé des difficultés à trouver la meilleur annotation de la relation. Nous avons décidé de laisser de ne pas séparer la conjonction de coordination dans l'argument. Par exemple, dans la phrase : *Plus tard , Han Sui et Ma Teng ont partagé une relation difficile l' un avec l' autre.* nous avons l'extraction suivante : ($A0$:Han Sui et Ma Teng, P :ont partagé une relation difficile, $A1$:l' un avec l' autre).

— Anaphore : Contrairement aux recommandations d'annotation, nous avons opté pour la non-résolution d'anaphores dans les extractions.

— Appositions : nous avons aussi décidé d'inclure l'extraction introduite par l'apposition alors que cette extraction peut être considéré comme redondante. Par exemple, dans la phrase : *La livraison de l' A321 s' est déroulée en présence de le pdg de la compagnie nationale Iranienne, Farhad Parvaresh* nous avons les deux extractions : ($A0$:La livraison de l' A321, P :s' est déroulée en présence de, $A1$:le pdg de la compagnie nationale Iranienne) et ($A0$:La livraison de l' A321, P :s' est déroulée en présence de, $A1$:Farhad Parvaresh).

Le référentiel d'évaluation final se compose de 506 phrases et 1783 relations.

4.4 Évaluations

Nous évaluons les différentes baselines en utilisant le framework d'évaluation proposé avec le référentiel d'évaluation standard CARB (Bhardwaj *et al.*, 2019). Nous rapportons le F1 et l'AUC (Area Under the Curve) score. Les systèmes de référence sont évaluées en exploitant le code de (Kolluru *et al.*, 2020).

4.5 Impact de l'architecture

Nous considérons une étude additionnelle pour étudier l'impact de notre nouvelle architecture, qui vise à séparer l'identification et l'étiquetage des arguments. Par conséquent, nous considérons comme système de référence le modèle *SpanOIE* présenté par (Zhan & Zhao, 2019). L'architecture de notre modèle est la même que l'architecture utilisé par (Zhan & Zhao, 2019) sauf que la notre se distingue en dissociant l'identification et la classification des arguments. En effet, notre architecture introduit une étape auxiliaire pour identifier les arguments du prédicat extrait avant d'étiqueter ces arguments,

1. https ://github.com/ssut/py-googletrans

tandis que (Zhan & Zhao, 2019) identifie et étiquette les arguments du prédicat extrait simultanément.

4.6 Systèmes de référence

Comme systèmes de référence, nous avons choisis le système à base de règle PredPatt(White *et al.*, 2016) et Multi2OIE(Ro *et al.*, 2020). Nous considérons deux variants pour notre modèle, le premier *FR-OIE* notre modèle entraîné en utilisant le corpus d'entraînement en français tandis que le deuxième *FR-OIE(En)* est le modèle entraîné sur le corpus d'entraînement original et qui est en anglais. Nous avons choisis d'inclure cet autre variant afin d'avoir une évaluation plus équitable et correcte avec Multi2OIE.

5 Résultats et discussion

Cette section présente les principales conclusions des résultats de l'expérience dans 5.1. Les résultats d'indépendance de domaine et de l'étude de l'impact de l'architecture sont discutés dans 5.2 et 5.3. Enfin, 5.4 présente une analyse des erreurs du modèle.

5.1 Performances

Les performances de chaque système par rapport au référentiel d'évaluation avec les différentes métriques sont rapportées dans le tableau 4. Les résultats de l'évaluation montrent que la méthode que nous proposons surpasse largement les autres baselines. Un autre résultat est que nous constatons est que le deuxième variant de notre modèle obtient de meilleurs résultats que Multi2OIE, qui est entraîné sur un corpus d'entraînement en anglais comme le deuxième variant de notre modèle. Ils semblent démontrer aussi que Multi2OIE basé sur BERT obtient des résultats légèrement meilleurs que l'approche à base de règles, PredPatt.

Système	Précision	Rappel	score F1	AUC
PredPatt (White *et al.*, 2016)	0.323	0.524	0.42	0.347
Multi2OIE (Ro *et al.*, 2020)	0.688	0.315	0.432	0.245
FR-OIE	0.727	0.627	**0.673**	0.496
FR-OIE(En)	0.702	0.596	**0.644**	0.461

TABLE 4 – Résultats d'évaluation des baselines et de notre approche sur le référentiel d'évaluation

5.2 L'indépendance de domaine

Afin de vérifier notre hypothèse, nous comparons aussi Multi2OIE et PredPatt sur un de nos corpus d'apprentissage issu de Wikipedia, le même domaine de donnée sur lequel a été pré-entraîné le modèle de langage BERT. Les résultats qui sont rapportés dans la table 5, montrent que contrairement au corpus issu du domaine de la finance, Multi2OIE dépasse largement PredPatt pour la précision et le score F1. Ces résultats montrent que les approches actuelles basées sur BERT ne supportent pas

Système	Précision	Rappel	F1	AUC
PredPatt (White *et al.*, 2016)	0.318	0.461	0.376	0.304
Multi2OIE (Ro *et al.*, 2020)	0.686	0.44	**0.536**	0.329

TABLE 5 – Résultats d'évaluation des baselines sur le corpus Wikipedia.

l'adaptation au domaine, qui est pourtant un critère essentiel dans l'OpenIE. Comme rapporté par (Li *et al.*, 2020), malgré leur habilité à extraire des représentations multilingues, les modèles de langage tel que BERT ne capturent que les caractéristiques spécifiques au domaine d'échantillon de données et n'extraient pas des caractéristiques indépendantes du domaine d'échantillon de données.

5.3 Résultats de l'étude de l'impact de l'architecture

L'architecture du modèle *FR-OIE(-Identification arguments)* correspond à celle utilisé par *SpanOIE* (Zhan & Zhao, 2019). Les résultats de l'impact de l'architecture rapportés dans 6, montrent que l'architecture proposée offre un gain de performance notable. Notre architecture proposée cible la performance du rappel, elle améliore la performance du rappel tout en entraînant une baisse de performance de la précision. Nous attribuons cela au fait de rechercher tous les arguments pertinents avant de les étiqueter à l'étape suivante est moins complexe et aboutit à un nombre plus important de relations prédicat-argument. Par conséquent, la performance du rappel augmente à mesure que le nombre de relations prédicat-argument augmente. Cependant, des relations prédicat-argument plus erronées seront propagées au module chargé d'étiqueter les arguments, ce dernier cherche uniquement à étiqueter les arguments extraits et ne peut pas rejeter ou détecter les arguments erronés, ce qui entraîne une baisse de performance de la précision.

Système	Précision	Rappel	F1	AUC
FR-OIE	0.727	**0.627**	**0.673**	**0.496**
FR-OIE(-Identification des arguments)	**0.746**	0.563	0.642	0.447

TABLE 6 – Résultats de l'étude d'impact de l'architecture

5.4 Analyse des erreurs

Comme prévu, la principale source d'erreurs était due aux erreurs de propagation de l'analyseur. Nous constatons que notre système échoue face aux constructions linguistiques complexes.
Le troisième exemple de 7 montre un exemple de *Gapping*, un type d'ellipse, où notre système échoue à extraire les relations correspondantes. La bibliothèque Stanza que nous avons utilisée n'arrive pas à détecter l'ellipse dans la phrase et alimente un arbre syntaxique incorrect dans notre modèle.
Une autre source d'erreur importante était le champ d'argument n-aire, où la relation n-aire était extraite en tant que relation binaire, avec l'argument n-aire manquant ou se trouvant dans le champ de l'objet. Le premier exemple de 7 montre un exemple dû à l'ambiguïté de l'attachement de préposition, où *en juillet 2010* est extrait dans le champ de l'objet *par la Hong Kong Monetary Authority*.
De plus, notre système échoue plus souvent à extraire des relations ayant comme prédicat des nominales, comme indiqué dans 7.
Enfin, le dernier exemple de 7 est relatif à au deuxième variant de notre modèle et qui est entraîné sur un corpus en anglais, il montre une construction linguistique spécifique au français (*objet indirect agentif* (exprimé par **iobj :agent** dans l'arbre syntaxique au format Universal Dependency) où l'agent initial (le pronom *lui* dans l'exemple) a été rétrogradé et est devenu un objet indirect. Puisque le deuxième variant de notre modèle a été entraîné sur un corpus en anglaises, il échouera naturellement face à des constructions spécifiques au français.

Type d'erreurs	Exemple
Arguments n-aire	Chinese Yuan Offshore (abréviation : CNH) monnaie chinoise lancée en juillet 2010 par la Hong Kong Monetary Authority (HKMA) **Extraction :** (**A0 :**Chinese Yuan Offshore **; P :**lancée **;** **A1 :**en juillet 2010 par la Hong Kong Monetary Authority) **Référence :** (**A0 :**Chinese Yuan Offshore **; P :**lancée **;** **A1 :**par la Hong Kong Monetary Authority **; A2 :**en juillet 2010)
Relations nominales	L'appétit de les entreprises pour la devise chinoise est bridé par les contrôles des capitaux. (**A0 :**les entreprises ; **P :**[ont un] appétit ; **A1 :**pour la devise chinoise)
Constructions linguistiques complexes ou ambiguës	L'objectif de cours de l' équipementier passe de 22 à 25,5 euros et celui du constructeur du Rafale de 1.100 à 1.260 euros. **Extraction :** (**A0 :**L'objectif de cours de l' équipementier **;** **P :**passe **; A1 :**de 22 **; A2 :**à 25,5 euros) **Référence :** (**A0 :**L'objectif de cours de l' équipementier **;** **P :**passe **; A1 :**de 22 **; A2 :**à 25,5 euros) **Référence** (**A0 :**L'objectif de cours du constructeur du Rafale **;** **P :**passe **; A1 :**de 1.100 **; A2 :**à 1.260 euros)
Constructions linguistiques propres au français	Google et Facebook en embuscade face à Apple, seul Google lui tient un peu tête. **Extraction :** (**A0 :**Google **; P :**tient un peu tête **;**) **Référence** (**A0 :**Google **; P :**tient un peu tête **; A1 :**lui)

TABLE 7 – Analyse et types des erreurs les plus récurrentes

6 Conclusion

Dans cet article, nous avons proposé une première approche d'OpenIE pour le Français, à base de réseaux de neurones. Notre modèle proposé introduit une étape auxiliaire pour identifier les arguments avant leur étiquetage et obtient les meilleurs résultats. Comme les référentiels d'évaluation actuels ne sont pas assez divers (Wikipedia et actualités), nous proposons un référentiel d'évaluation issu du domaine de la finance pour évaluer la performance et la robustesse des systèmes d'une manière plus précise. Le code du prototype ainsi que les données sont disponibles dans le répertoire du projet. [2]

Références

BALASUBRAMANIAN N., SODERLAND S., ETZIONI O. *et al.* (2013). Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 1721–1731.

BHARDWAJ S., AGGARWAL S. & MAUSAM M. (2019). CaRB : A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 6262–6267, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1651.

CHRISTENSEN J., MAUSAM, SODERLAND S. & ETZIONI O. (2010). Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, p. 52–60, Los Angeles, California : Association for Computational Linguistics.

CORRO L. D. & GEMULLA R. (2013). Clausie : clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, p. 355–366.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DOZAT T. & MANNING C. D. (2016). Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv :1611.01734*.

FADER A., SODERLAND S. & ETZIONI O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 1535–1545, Edinburgh, Scotland, UK. : Association for Computational Linguistics.

FAN A., GARDENT C., BRAUD C. & BORDES A. (2019). Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv preprint arXiv :1910.08435*.

FORNEY G. (1973). The viterbi algorithm. *Proceedings of the IEEE*, **61**(3), 268–278. DOI : 10.1109/PROC.1973.9030.

GAMALLO P. & GARCIA M. (2015). Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, p. 711–722 : Springer.

2. https://github.com/atmani-massinissa/UD2OIE

GOTTI F. & LANGLAIS P. (2016). From french wikipedia to erudit : A test case for cross-domain open information extraction. *Computational Intelligence*. DOI : 10.1111/coin.12120.

JABBARI A., SAUVAGE O., ZEINE H. & CHERGUI H. (2020). A French corpus and annotation schema for named entity recognition and relation extraction of financial news. In *Proceedings of the 12th Language Resources and Evaluation Conference*, p. 2293–2299, Marseille, France : European Language Resources Association.

KOLLURU K., ADLAKHA V., AGGARWAL S., MAUSAM & CHAKRABARTI S. (2020). OpenIE6 : Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 3748–3761, Online : Association for Computational Linguistics.

LAFFERTY J., MCCALLUM A. & PEREIRA F. C. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, p. 282–289.

LECHELLE W., GOTTI F. & LANGLAIS P. (2019). WiRe57 : A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, p. 6–15, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-4002.

LI J., HE R., YE H., NG H. T., BING L. & YAN R. (2020). Unsupervised domain adaptation of a pretrained cross-lingual language model. *arXiv preprint arXiv :2011.11499*.

MAUSAM M. (2016). Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, p. 4074–4077.

NIVRE J., DE MARNEFFE M.-C., GINTER F., GOLDBERG Y., HAJIČ J., MANNING C. D., MCDONALD R., PETROV S., PYYSALO S., SILVEIRA N., TSARFATY R. & ZEMAN D. (2016). Universal Dependencies v1 : A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, p. 1659–1666, Portorož, Slovenia : European Language Resources Association (ELRA).

PAL H. & MAUSAM (2016). Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, p. 35–39, San Diego, CA : Association for Computational Linguistics. DOI : 10.18653/v1/W16-1307.

QI P., ZHANG Y., ZHANG Y., BOLTON J. & MANNING C. D. (2020). Stanza : A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, p. 101–108, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-demos.14.

RO Y., LEE Y. & KANG P. (2020). Multi^2OIE : Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 1107–1117, Online : Association for Computational Linguistics.

STANOVSKY G., DAGAN I. *et al.* (2015). Open ie as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, p. 303–308.

STANOVSKY G., MICHAEL J., ZETTLEMOYER L. & DAGAN I. (2018). Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 885–895, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1081.

WHITE A. S., REISINGER D., SAKAGUCHI K., VIEIRA T., ZHANG S., RUDINGER R., RAWLINS K. & VAN DURME B. (2016). Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1713–1723, Austin, Texas : Association for Computational Linguistics. DOI : 10.18653/v1/D16-1177.

YAHYA M., WHANG S., GUPTA R. & HALEVY A. (2014). ReNoun : Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 325–335, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1038.

YAO H., ZHU D.-L., JIANG B. & YU P. (2019). Negative log likelihood ratio loss for deep neural network classification. In *Proceedings of the Future Technologies Conference*, p. 276–282 : Springer.

YATES A., BANKO M., BROADHEAD M., CAFARELLA M., ETZIONI O. & SODERLAND S. (2007). TextRunner : Open information extraction on the web. In *Proceedings of Human Language Technologies : The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, p. 25–26, Rochester, New York, USA : Association for Computational Linguistics.

ZHAN J. & ZHAO H. (2019). Span based open information extraction. *arXiv preprint arXiv :1901.10879*.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Plongements Interprétables pour la Détection de Biais Cachés

Tom Bourgeade[1] Philippe Muller[1] Tim Van de Cruys[2]
(1) IRIT, Université Toulouse 3, 31062 Toulouse, France
(2) Leuven.AI institute, KU Leuven, 3000 Louvain, Belgique
tom.bourgeade@irit.fr, muller@irit.fr, tim.vandecruys@kuleuven.be

RÉSUMÉ

De nombreuses tâches sémantiques en TAL font usage de données collectées de manière semi-automatique, ce qui est souvent source d'artefacts indésirables qui peuvent affecter négativement les modèles entraînés sur celles-ci. Avec l'évolution plus récente vers des modèles à usage générique pré-entraînés plus complexes, et moins interprétables, ces biais peuvent conduire à l'intégration de corrélations indésirables dans des applications utilisateurs. Récemment, quelques méthodes ont été proposées pour entraîner des plongements de mots avec une meilleure interprétabilité. Nous proposons une méthode simple qui exploite ces représentations pour détecter de manière préventive des corrélations lexicales faciles à apprendre, dans divers jeux de données. Nous évaluons à cette fin quelques modèles de plongements interprétables populaires pour l'anglais, en utilisant à la fois une évaluation intrinsèque, et un ensemble de tâches sémantiques en aval, et nous utilisons la qualité interprétable des plongements afin de diagnostiquer des biais potentiels dans les jeux de données associés.

ABSTRACT

Interpretable Embeddings for Hidden Biases Detection

A lot of current semantic NLP tasks use semi-automatically collected data, that are often prone to unwanted artifacts, which may negatively affect models trained on them. With the more recent shift towards more complex, and less interpretable, pre-trained general purpose models, these biases may lead to undesirable correlations getting integrated into end-user applications. Recently a few methods have been proposed to train word embeddings with better interpretability. We propose a simple setup which exploits these representations to preemptively detect easy-to-learn lexical correlations in various datasets. We evaluate a few popular interpretable embedding models for English for this purpose, using both an intrinsic evaluation, and a large set of downstream semantic tasks, and we make use of the embeddings' interpretable quality in order to diagnose potential biases in the associated datasets.

MOTS-CLÉS : Interprétabilité, Plongements lexicaux, Biais.

KEYWORDS: Interpretability, Word embeddings, Bias.

1 Introduction

Les modèles de plongements de mots sont une méthode populaire et efficace pour l'association de tokens linguistiques à des représentations vectorielles, qui peuvent ensuite être exploitées par des architectures de réseaux de neurones dans le cadre de tâches diverses en traitement automatique des langues (TAL). Les modèles de plongements denses, tels que word2vec (Mikolov *et al.*, 2013), GloVe (Pennington *et al.*, 2014), ou fastText (Bojanowski *et al.*, 2017), font correspondre les mots de leur vocabulaire à des vecteurs denses de quelques centaines de dimensions (généralement 300 ou 500),

dérivés de manière non supervisée (ou auto-supervisée) de statistiques de cooccurrences extraites de grands corpus textuels, et existent pour de nombreuses langues. Ce type de modèles permettent d'obtenir de très bons résultats sur de nombreuses tâches de TAL en aval. Souvent, un simple calcul de moyenne (Arora *et al.*, 2017) ou d'addition matricielle (Kober *et al.*, 2017) des représentations de plusieurs mots peut donner des représentations efficaces des phrases, qui peuvent être directement exploitées par des modèles de classification tout aussi simples - fréquemment avec des performances étonnamment bonnes. Ces résultats indiquent que les modèles de plongements de mots denses ont tendance à capturer les informations sémantiques dans les énoncés en langage naturel. Cependant, le manque d'interprétabilité est un problème important pour la majorité de ces modèles, car il est pratiquement impossible de caractériser qualitativement la sémantique des différentes dimensions de la représentation d'un mot (voir par exemple fastText dans la Table 1).

Les notions d'interprétabilité et d'explicabilité sont difficiles à définir en tant que telles en règle général, mais ici nous nous intéressons en particulier à ces notions comme outils de détection et de caractérisation de biais dans des jeux de données. Cette notion de biais peut être associée à la notion concrète de décalage distributionnel des labels (Torralba & Efros, 2011; He *et al.*, 2019) entre un ensemble d'entraînement et un ensemble de test (à condition que ce dernier soit bien choisi, en particulier pour mettre en exergue cette notion, ce qui n'est malheureusement pas toujours le cas).

Dans le cas des modèles de plongements denses, la difficulté de comprendre comment les valeurs de chaque dimension se traduisent en informations sémantiques encodées se propage aux modèles de TAL exploitant ces plongements, excluant donc la possibilité d'expliquer directement, de manière numérique, le comportement d'un modèle, même linéaire, en fonction de ces dimensions – les méthodes d'explicabilité *a posteriori* en TAL emploient de ce fait le plus souvent une analyse en fonction de la présence ou non de vecteur-mots entiers, comme par exemple dans Li *et al.* (2016) ou Ribeiro *et al.* (2016) – ce qui rend difficile la détection de biais lexicaux cachés. Une alternative existe cependant, sous la forme de modèles de plongements interprétables, dont les dimensions, par construction, sont plus aptes à fournir des indications sur les champs lexicaux associés aux mots encodés.

Ces représentations auraient pour avantages de permettre des analyses de corrélations lexicales à des niveaux plus abstraits (liés aux aspects sémantiques encodés dans ces plongements) et donc plus facilement exploitables de manière générale, que la simple présence ou non de mots particuliers dans les entrées d'un modèle par exemple.

Une façon d'utiliser ces plongements interprétables dans la pratique serait de détecter et de réparer les biais ou les artefacts d'annotation/construction éventuellement présents dans les ensembles de données, appris de façon indésirable par les modèles de prédiction formés sur eux. Bien que beaucoup de travaux ont été réalisés pour isoler et corriger ces problèmes, les méthodes existantes exigent presque toujours une connaissance préalable de leur présence et de leur nature, et celle-ci est souvent acquise après une analyse qualitative du comportement de modèles suspects sur des tâches en aval, parfois des années après les faits (par exemple, l'ensemble de données SNLI, voir la section 2). Notre objectif est de détecter ces problèmes avant qu'ils ne s'infiltrent et n'entachent les résultats des recherches ultérieures ou les applications utilisateurs finales.

Nous proposons donc une approche simple pour diagnostiquer qualitativement les problèmes potentiels dans les ensembles de données de TAL. L'idée clé est d'exploiter les caractéristiques des modèles de plongements de mots interprétables existants comme indices de biais éventuels présents dans les données : nous entraînons un classifieur CBOW (*Continuous Bag-Of-Words*) intentionnellement élémentaire et de ce fait fondamentalement interprétable (par analyse direct des paramètres appris),

qui utilise simplement la moyenne des représentations des mots d'un texte comme caractéristiques d'entrée, qui sont ensuite introduites dans une couche de régression *Softmax*. En analysant les performances et les paramètres appris par cette couche de classification, qui correspondent chacun à une dimension interprétable dans la matrice de plongements choisie, nous sommes en mesure d'obtenir des indications sur d'éventuels biais lexicaux faciles à apprendre et à exploiter, dû à la nature élémentaire du modèle en question, et de déterminer qualitativement s'ils sont attendus ou non.

Nos principales contributions sont les suivantes : (1) une nouvelle variante d'une méthode d'évaluation intrinsèques de détection d'intrusion de mots (*word intrusion detection*), appliquée à divers modèles de plongements interprétables populaires en anglais, suivie par, (2) une évaluation extrinsèque de ces mêmes modèles par rapport à un ensemble de tâches de TAL en aval qui pourraient potentiellement mettre en évidence des artefacts d'annotation intéressants, et (3) une analyse de certains classifieurs produits qui démontrent le potentiel de cette approche pour analyser des ensembles de données.

2 Travaux Connexes

Un certain nombre d'approches ont été proposées pour l'induction de plongements interprétables, qui peuvent être divisés en deux catégories différentes : les modèles de plongements fondés sur des contraintes, et les modèles enrichis avec des informations *a priori*. La majorité des modèles de la première catégorie se concentrent sur deux types de contraintes qui améliorent l'interprétabilité des vecteurs de plongements des mots : la parcimonie et la non-négativité. Un large éventail de contributions (Lee & Seung, 1999; Fyshe *et al.*, 2014; Faruqui *et al.*, 2015; Dahiya *et al.*, 2016; Trifonov *et al.*, 2018; Subramanian *et al.*, 2018) ont montré que ces deux propriétés améliorent considérablement la capacité à comprendre à quoi correspond chaque dimension dans une représentation de mot en termes de sémantique abstraite. La parcimonie signifie que le nombre de dimensions différentes par lesquels un mot peut être encodé est limité, ce qui permet de les répertorier et de les analyser de manière relativement exhaustive. La non-négativité quant à elle signifie que les valeurs de chaque dimension peuvent être interprétées comme une "participation" relative de la sémantique associé dans la représentation. Les plongements peuvent être créés en imposant des contraintes à la méthode de construction (Murphy *et al.*, 2012; Panigrahi *et al.*, 2019), ou les contraintes peuvent être imposées comme étape supplémentaire *a posteriori*, en transformant les vecteurs de plongements de mots denses standards en représentations de facto plus éparses et moins bruyantes, par exemple par l'utilisation d'algorithmes de rotation de base connus de l'analyse en composantes principales et de l'analyse factorielle (Park *et al.*, 2017; Dufter & Schütze, 2019), ou par la factorisation en matrices non négatives (Faruqui *et al.*, 2015; Subramanian *et al.*, 2018).

Une seconde avenue de recherche tente d'injecter des informations sémantiques *a priori* dans les modèles de plongements afin d'améliorer leur interprétabilité : Hurtado Bodell *et al.* (2019) utilisent des informations préalables, sous la forme de paires de mots censés être discriminés par une dimension particulière (*homme-femme* pour une dimension de genre par exemple), afin de guider les plongements appris vers des formes plus facilement interprétables ; Fyshe *et al.* (2014) incorporent des données d'activation cérébrale – recueillies auprès des participants pendant qu'ils lisent les mots associés – dans un modèle de plongements de mots interprétables basé sur des contraintes, *Non-Negative Sparse Embedding* (NNSE) (Murphy *et al.*, 2012). Nous nous concentrons ici sur le premier type d'approches de représentations interprétables, les plongements construits sous contraintes, car elles peuvent être employées plus facilement dans des contextes similaires à ceux pour lesquels des modèles de plongements denses sont plus habituellement employés.

Ces dernières années, un certain nombre d'artefacts statistiques indésirables ont été découverts

fastText	NMF300
tortricidae, baronetage, poaceae, prószyński	desktop, server, linux, microsoft, firmware
eum, cydia, inj, papaya, honeydew	leaved, eucalyptus, trees, planted, juniper
kapamilya, inkigayo, noosa, pvo, puso	flavored, dessert, drinks, chocolate, drink
NNSE	**SPOWV**
mango, raspberry, peach, lemon, pear	onion, sauce, pradesh, streak, salad
strawberries, peaches, oranges, pears, apples	scout, scouts, fellows, dry, cub
fir, birch, pine, willow, spruce	malayalam, leopard, grape, karnataka, raft
SPINE	**Word2Sense**
grape, wine, wines, vineyards, winery	ipod, iphone, apple, mini, lansing
linux, windows, macintosh, playstation, xbox	macintosh, intel, apple, mac, dell
bread, lime, dessert, 1/4, apples	apples, citrus, fruits, ripe, berries

TABLE 1 – Comparatif qualitatif des dimensions des différents modèles de plongements de mots utilisés ici : sont listés ici les 5 mots associés avec la valeur la plus grande dans les 3 dimensions les plus actives pour le mot “apple”. On remarque que le modèle dense fastText n’est manifestement pas interprétable dans ce sens, tandis que différentes sémantiques du mot (y compris celles liées à la société informatique homonyme) ont été capturées par les modèles interprétables.

dans des ensembles de données de TAL bien connus et largement utilisés, par exemple dans la tâche d’inférence en langage naturel (NLI), en particulier dans le jeu de données SNLI (Stanford Natural Language Inference) introduit par Bowman *et al.* (2015) ainsi que dans sa variante améliorée multi-genres MNLI (Williams *et al.*, 2018), pour lesquelles Gururangan *et al.* (2018) et Poliak *et al.* (2018) ont par exemple découvert que des modèles “hypothèse-seule”, qui ne reçoivent qu’une partie de l’entrée, peuvent correctement prédire les étiquettes de grandes parties de ces corpus, ce qui indique la présence de corrélations indésirables, causées en partie par l’annotation de données par myriadisation. McCoy *et al.* (2019) montrent également que ces modèles ont tendance à reposer sur des heuristiques ad hoc “faciles” (quantité de chevauchements de mots, par exemple), elles aussi indicatives de problèmes dans les données d’apprentissage pour cette tâche. Afin de surmonter ces problèmes, He *et al.* (2019) conçoivent une méthode pour entraîner des modèles débiaisés sur ces corpus ; pour ce faire, ils entraînent d’abord un modèle biaisé qui exploite principalement les artefacts indésirables (en ne lui fournissant que des informations incomplètes, comme dans le cas des modèles à hypothèse-seule), et entraînent ensuite un nouveau modèle théoriquement débiaisé sur les résidus (dans le sens des instances avec une faible erreur) du classifieur biaisé obtenu précédemment.

3 Plongements Interprétables

Nous nous sommes concentrés ici sur quatre différents modèles de plongements interprétables non-négatifs et parcimonieux que l’on peut trouver dans la littérature pertinente, avec des niveaux de complexité variables :

- NNSE [1] (Murphy *et al.*, 2012), construit par reconstruction avec erreur quadratique modifiée de statistiques de concurrences (inspiré de la méthode *Non-Negative Sparse Coding* de Hoyer (2002)) collectées sur le jeu de données web anglais ClueWeb09 ;

1. http://www.cs.cmu.edu/~bmurphy/NNSE/

- SPOWV [2] (Faruqui *et al.*, 2015) et SPINE [2] (Subramanian *et al.*, 2018), tout deux construits directement sur un modèle de plongements denses existant (ici, nous utilisons les versions calculées sur GloVe), le premier par factorisation matricielle, et le second à l'aide d'un auto-encodeur k-éparse ;
- Word2Sense [3] (Panigrahi *et al.*, 2019), construit à l'aide d'une approche fondée sur l'allocation de Dirichlet latente, sur une combinaison des jeux de données UKWAC et Wackypedia ;

A ces modèles existants, nous avons ajouté nos propres plongements interprétables non-négatifs parcimonieux, NMF300, construits simplement par factorisation en matrices non-négatives (avec une largeur de 300 dimensions) à l'aide des règles de mise-à-jour multiplicatives définies par Lee & Seung (2001) pour la métrique de divergence de Kullback-Leibler, sur des statistiques de cooccurrences issues d'articles Wikipedia. Des cinq modèles étudiés ici, celui-ci est le plus simple, dans le sens où la seule contrainte présente lors de la factorisation est la non-négativité, nous l'employons comme une sorte de modèle étalon comparatif.

Tous les modèles de plongements utilisés ici sont non-contextualisés et statiques par nature, cependant, comme le montre Kober *et al.* (2017), l'utilisation d'une simple opération de composition (addition matricielle ou moyennage, par exemple) sur la séquence de représentations vectorielles correspondant à une phrase permet en pratique de désambiguïser contextuellement les différents sens encodés dans les plongements de mots. De plus, tous ces modèles ayant été originalement construits dans des contextes très différents (taille des plongements : 300 pour NNSE, 1000 pour SPOWV et SPINE, et 2250 pour Word2Sense ; tailles des vocabulaires ; corpus sources ; etc.), et en accord avec la tendance actuelle de mener des expériences avec des modèles pré-entraînés, nous avons fait le choix d'utiliser les versions fournies par leurs auteurs respectifs, même si de ce fait les paramètres de leur construction sont assez hétérogènes. Cette hétérogénéité est difficile à contrôler puisqu'un réglage fin de ces paramètres aurait été nécessaire à réaliser indépendamment pour chaque approche même sur un jeu de données d'entraînement unifié.

4 Jeux de Données Utilisés

Nous présentons ici rapidement les jeux de données annotées ou collectées que nous avons analysés pour détecter d'éventuels biais indésirables. Ils représentent diverses tâches de classification simple ou de relations, couvrant différents aspects sémantiques et genres textuels :

- **IMDB** (Maas *et al.*, 2011) est une collection de critiques de films (petits paragraphes) recueillies sur le site web IMDB, annotées avec des étiquettes de sentiments binaires (positif/négatif), dérivées des scores des critiques.
- **BoolQ** (Clark *et al.*, 2019) est un jeu de données de questions oui/non "d'origine naturelle", issues de requêtes sur un moteur de recherche, associées à des passages textuels issues d'articles Wikipedia pertinents permettant normalement de répondre à la question.
- **Sarcasm** (Oraby *et al.*, 2016) est un recueil d'extraits de débats de forums en ligne, composé d'une déclaration et d'une réponse, qualifiées de sarcastiques ou non.
- **UR-FUNNY** (Hasan *et al.*, 2019) est un ensemble de données multimodales créé pour soutenir l'analyse de l'humour, en intégrant le langage naturel, la parole et la vidéo. Nous utilisons ici la partie textuelle seule, composée d'instances de paires d'un texte contexte et d'une phrase de chute, étiquetées comme humoristiques ou non.

2. https://github.com/harsh19/SPINE#word-embeddings
3. https://github.com/abhishekpanigrahi1996/Word2Sense#pretrained-vectors

- **SST** (Socher *et al.*, 2013) (Stanford Sentiment Treebank) est un recueil de phrases tirées de critiques de films, annotées pour la polarité des sentiments au niveau des phrases des arbres de syntaxe. Nous utilisons ici l'annotation en cinq classes de haut niveau fournie, ramenée aux trois classes "positive", "négative" et "neutre" (en fusionnant les classes originales "très positive/négative" avec leur équivalent correspondant).
- **SNLI** (Bowman *et al.*, 2015) est une collection de paires de phrases ou de descriptions textuelles, conçue pour tester la capacité des modèles à prédire les relations inférentielles. Les étiquettes possibles pour la relation sont "inférence" (*entailment*), "contradiction" ou "neutre".
- **Emergent** (Ferreira & Vlachos, 2016) est un jeux de données pour la classification du positionnement, où chaque instance est composée d'une affirmation et de titres d'articles de journaux portant sur cette affirmation, avec trois labels de positionnement possible : "pour" (*for*) si les titres supportent l'affirmation, "observe" (*observing*) si les titres n'affichent pas de prise de position clair, ou "contre" (*against*) s'il contredisent l'affirmation.
- **PDTB** (Prasad *et al.*, 2008) (Penn Discourse TreeBank) est une partie du corpus Penn TreeBank, annoté de relations *rhétoriques*, soit entre les clauses d'une phrase, soit entre des phrases voisines. Nous utilisons ici seulement les 11 classes présentes dans le jeu de test (le jeu d'entraînement en contient normalement 16), ce qui est la pratique courante.

Nous répertorions également les caractéristiques de ces jeux de données dans la Table 2.

Corpus	E	T	C	Equilibrage/Classes principles
IMDB	25000	22500	2	eq.
BoolQ	9427	2943	2	true=62.3%, false=37.7%
Sarcasm	3754	469	2	eq.
UR-FUNNY	8074	1058	2	eq.
SST	8544	1989	3	positive=42.0%, negative=39.2%, neutral=18.8%
SNLI	549367	9824	3	entailment=33.4%, contradiction=33.3%, neutral=33.3%
Emergent	2076	259	3	for=47.7%, observing=37.0%, against=15.3%
PDTB	12907	1085	11	cause=26.5%, conjunction=22.1%, restatement=19.1%, contrast=12.4%, *reste*=19.9%

TABLE 2 – Statistiques des différents corpus utilisés : E/T = Taille des jeux d'Entraînement/Test respectivement ; C = Nombre de Classes. Pour PDTB, seules les 4 classes majoritaires sont répertoriées (ce corpus présente un fort déséquilibre entre les classes, notamment dans l'ensemble de test, où certaines classes ne sont pas du tout représentées).

5 Expériences et Résultats

5.1 Détection d'Intrusion de Mots

Afin d'évaluer qualitativement l'interprétabilité des différents modèles de plongements sous contraintes explorés ici, nous modifions la méthode d'évaluation de la détection d'intrusion de mots introduite dans Chang *et al.* (2009), qui semble être devenue la norme de facto à cet effet au fil des ans (Murphy *et al.*, 2012; Fyshe *et al.*, 2014; Faruqui *et al.*, 2015; Subramanian *et al.*, 2018). On peut la résumer ainsi : étant donné un échantillon mélangé de mots (généralement 4 ou 5) choisis

parmi les mots les plus “actifs” pour une dimension d’une matrice de plongements interprétables (dans le cas d’une matrice non négative, les mots ayant les plus grandes valeurs dans cette dimension particulière), auquel est ajouté un mot “intrus”, choisi parmi les mots les moins actifs pour cette dimension, un évaluateur humain peut-il trouver l’intrus ?

Nous avons ici employé une variante plus difficile de cette méthode, compte-tenu de la variété de modèles explorés, sur un échantillon de 50 dimensions pour chaque matrice de plongements (ces dimensions correspondant à la dimension la plus active pour 50 mots tirés dans l’intersection des vocabulaires des 5 modèles évalués ici), avec 3 évaluateurs (auteurs de cet article, en aveugle). Dans la variante “classique” de cette tâche, le mot intrus est généralement choisi parmi les mots les moins actifs de la dimension évaluée, et parmi les mots plus actifs d’une autre dimension (le plus souvent choisi aléatoirement). Après des essais sur de petits échantillons d’instances produites par cette première approche, nous avons remarquer des difficultés à différencier les différents modèles selon les performances obtenues, car la tâche semble trop “facile” pour la plupart des modèles interprétables (à l’exception de `SPOWV`). De plus, conceptuellement, choisir la dimension fortement active du mot intrus aléatoirement ne permet pas vraiment de distinguer la caractéristiques discriminante de la dimensions étudiée : idéalement, il faudrait pouvoir évaluer la dimension cible indépendamment des autres dimensions actives potentiellement communes aux “vrais” mots. Pour ce faire, nous avons modifier le processus de sélection de l’intrus : celui-ci est similairement choisi parmi les 10% des mots les moins actifs de la dimension cible, mais avec comme second critère le fait d’avoir pour seconde dimension la plus active la seconde dimension la plus active “commune” aux vrais mots (après expérimentation, nous sélectionnons celle avec la plus grande médiane parmi les vrais mots). Qualitativement, cela a pour effet d’augmenter significativement la difficulté à repérer l’intrus, notamment quand la dimension cible est plus ou moins associée à un champ lexical peu spécifique.

Voici un exemple d’application de cette méthode pour la dimension cible n°110 du modèle `SPINE` pré-entraîné utilisé ici : les 10 mots les plus actifs de cette dimension sont en ordre décroissant *“pius, pope, diocese, bishops, basilica, archdiocese, benedict, vatican, catholic, bishop”*, les 4 premiers mots les plus actifs sont donc sélectionnés comme vrais mots. Leur seconde dimension commune la plus active (au sens de la seconde médiane la plus élevée des valeurs de leurs dimensions) est la dimension n°178, avec pour mots les plus actifs *“baptist, jesus, christians, holy, lutheran, religious, judaism, believers, prayers, baptism”*, qui visiblement semble couvrir des champs lexicaux proches de ceux de la dimension cible. Dans la version “classique” de la tâche, on sélectionnerait ici un intrus aléatoirement parmi les 10% des mots les moins actifs de la dimension cible, dont *“baseline, sculptures, feedback, armoured, modeled”* serait un échantillon de 5 mots. Dans la variante plus difficile présentée ici, un même échantillon de 5 mots serait de plus tiré parmi les mots les plus actifs dans la dimension n°178 dans cette proportion, ici *“judaism, mormon, preacher, buddhism, meditation”*. On peut qualitativement observer que ce second échantillon est plus proche des 4 vrais mots *“pius, pope, diocese, bishops”* que l’échantillon aléatoire, mais en même temps semble démontrer les spécificités uniques de ces deux dimensions (la dimension n°110 semble ici plus spécifique aux termes associés à la religion catholique, tandis que la n°178 semble correspondre à des termes religieux plus généraux).

Nous n’avons pas effectué cette évaluation complète sur un modèle de plongements denses car après essai sur un échantillon, la précision des évaluateurs était équivalente à un choix aléatoire (traduisant la non-interprétabilité de ces modèles), ce qui apparaît également dans les résultats de Subramanian *et al.* (2018) par exemple.

Exemple d'instance la tâche : *"dermatologist, columnist, veterinarian, psychiatrist, pathologist"* est un ensemble de mots pour une dimension particulière de `NNSE`, où *"columnist"* est l'intrus ici.

Modèle	Précision moyenne évaluateurs	Accord Inter-évaluateurs	Kappa de Fleiss
`NMF300`	76%	**94%** ; 72%	0.74
`NNSE`	**79%**	90% ; **74%**	**0.76**
`SPOWV`	38%	84% ; 34%	0.43
`SPINE`	**79%**	92% ; 60%	0.63
`Word2Sense`	65%	88% ; 56%	0.61

TABLE 3 – Résultats de l'évaluation de la Détection d'Intrusion de Mots effectuée sur les 5 modèles considérés (L'accord est donné sous la forme : `majorité ; unanimité`).

Nous notons tout d'abord que nos résultats (Table 3) sont à peu près similaires à ceux de Subramanian *et al.* (2018) et Panigrahi *et al.* (2019) pour les modèles `SPOWV`, `SPINE` et `Word2Sense`, compte tenu des légères différences dans le dispositif expérimental, et de la nature intrinsèquement subjective de l'évaluation. Si les performances globales de tous les modèles sont assez bonnes, nous constatons après une analyse qualitative des dimensions des modèles que leur interprétabilité est très hétérogène : si la majorité est relativement facile à associer à des champs lexicaux précis, certaines semblent capter des phénomènes pseudo-lexicaux, selon les corpus dont ils sont issus. Par exemple, les modèles entraînés sur les articles de Wikipedia peuvent être la proie de certains artefacts de fréquences causés par des données tabulaires très répétitives, ou, plus subtilement, par des ensembles d'articles appartenant aux mêmes domaines. Par exemple, des articles sur un sport particulier conduisent à des articles sur des équipes sportives particulières, qui conduisent à des articles sur des joueurs sportifs particuliers, etc, occupant une place dans le corpus disproportionnée.

5.2 Évaluation sur les Tâches en Aval

Nous avons également choisi d'évaluer l'interprétabilité et l'utilité des plongements en les employant dans des tâches de classification de texte, et en vérifiant si certaines dimensions jouaient un rôle important dans les prédictions, mettant éventuellement au jour des corrélations lexicales indésirables faciles à apprendre. Pour chaque tâche, nous fabriquons donc un classifieur de régression *Softmax* élémentaire (avec une matrice de paramètres de taille $|dimensions\ des\ plongements| \times |classes|$, ainsi que les biais), prenant en entrée la moyenne des plongements interprétables des mots de la phrase, ou, dans le cas des tâches avec deux phrases en entrée, la concaténation vectorielle suivante (inspirée de Conneau *et al.* (2017)) : $< u, v, |u - v|, u * v >$ (u/v : moyenne des vecteurs première/deuxième phrase ; $|x|$: valeur absolue terme à terme du vecteur x ; $*$: opérateur de produit terme-à-terme). Pour chaque couple $(\text{modèle}, \text{corpus})$, nous entraînons le classifieur approprié pour un maximum de 200 époques en utilisant l'optimiseur Adam, avec 50 époques antérieures de réglages fins automatiques des hyperparamètres, en utilisant l'algorithme *Tree-structured Parzen Estimator* (Bergstra *et al.*, 2013, 2011), via son implémentation par la bibliothèque `optuna` (Akiba *et al.*, 2019).

Nous évaluons ensuite chaque classifieur produit sur son jeu de test respectif, et affichons les performances globales pour le corpus dans la Table 4. En plus des cinq modèles de plongements interprétables, nous avons également entraîné de la même manière un ensemble de classifieurs en utilisant les plongements en anglais dense `fastText` (Bojanowski *et al.*, 2017) (sans information

sur les sous-mots)[4], pour comparer l'efficacité des modèles interprétables contraints par rapport aux modèles denses, dans cette configuration de classification élémentaire. Nous présentons également les résultats du classifieur de base "factice" (Dummy) équivalent, qui génère des prédictions au hasard, en suivant la distribution des classes sur l'ensemble de test pour chaque corpus.

Modèle \ Corpus	IMDB	BoolQ	Sarcasm	UR-FUNNY	SST	SNLI	Emergent	PDTB
NMF300	67.8	62.6	60.5	57.7	54.6	58.6	50.9	33.2
NNSE	78.7	63.6	63.9	59.9	60.6	56.3	66.8	31.1
SPOWV	81.9	**66.9**	**70.5**	**65.0**	62.9	62.9	**72.2**	**36.6**
SPINE	81.3	65.9	67.8	63.6	59.9	64.1	**72.2**	34.5
Word2Sense	**82.2**	66.2	67.3	63.9	61.4	**65.5**	69.8	34.2
Dummy *(baseline)*	50.5	53.5	53.0	52.5	39.5	33.6	41.3	19.3
fastText	82.0	63.7	70.1	64.5	**64.4**	61.3	69.5	33.4
*Modèles Dédiés**	96.8	76.9	74†	64.4	96	91.5	73	48

TABLE 4 – Résultats de l'approche sur les tâches de classification en aval évalués. Les scores de justesse pour chaque paire modèle-corpus sont indiqués en pourcentages (les meilleurs scores de l'expérience sont en **gras**). *Nous énumérons également les résultats des modèles populaires dédiés à ces tâches, que l'on retrouve dans la littérature et qui atteignent (ou s'approchent) des performances de l'état de l'art, à titre de comparaison (IMDB, SST : Yang *et al.* (2019) ; BoolQ : Clark *et al.* (2019) ; Sarcasm : Oraby *et al.* (2016) ; UR-FUNNY : Hasan *et al.* (2019) ; SNLI : Liu *et al.* (2019) ; Emergent : Ferreira & Vlachos (2016) ; PDTB : Dai & Huang (2019)). †Mesure F1 pour la classe positive (justesse non disponible).

Nous constatons que, de manière assez surprenante, les classifieurs élémentaires entraînés avec des plongements interprétables semblent aussi performants, voire légèrement meilleurs, que leurs équivalents entraînés avec un modèle dense comme fastText. Il semble également que, pour la plupart des tâches étudiées ici, l'approche fonctionne assez bien, compte tenu de la nature simpliste des modèles de classification. Cela semble indiquer que de nombreuses tâches de TAL ont une composante purement lexicale plus ou moins forte, qui peut expliquer des sous-ensembles parfois importants des corpus correspondants, ce qui nous semble à un certain degré être problématique. En effet, s'il est cohérent que certains termes et champs lexicaux soient associés par nature, par exemple, à une classe de sentiment positive ou négative, la non-nécessité de prendre en compte les phénomènes structurels importants comme la négation (ce qui est le cas pour les modèles élémentaires utilisés ici, mais aussi pour une grande variété de modèles même plus complexes, comme le montre par exemple Naik *et al.* (2018), Kassner & Schütze (2020), ou Hossain *et al.* (2020), par construction d'instances où la compréhension de la négation est essentielle pour la classification, sur lesquelles la plupart des modèles de langue de l'état de l'art échouent) pour classifier une partie significative des données semblent indiquer un problème d'adéquation entre celles-ci et la tâche de TAL associée. Bien que ces résultats ne puissent à eux seuls indiquer l'existence de corrélations nécessairement erronées dans les données, les performances relativement importantes obtenus avec ces approches simples pourraient être le signe que certains biais indésirables entachent les ensembles de données.

Globalement, les modèles SPOWV, SPINE et Word2Sense semble être les plus performants, contrai-

4. wiki-news-300d-1M — https://fasttext.cc/docs/en/english-vectors.html

rement aux modèles plus simples, NMF300 et NNSE, avec quelques instances (BoolQ et SST pour le premier) de classes mal voir non-apprises (voir résultats des expériences [5]).

5.3 Diagnostics de Corpus

Dans cette section, nous présentons quelques éléments d'analyse qualitative des modèles entraînés, des fichiers contenant les résultats des évaluations sur les tâches en aval pour chaque modèle, ainsi que les dimensions les plus prédictives pour chaque classe de ces tâches étant à disposition [5] pour plus de détail.

D	M	C	I	P	Mots les plus actifs dans la dimension
IMDB	NNSE	*pos*	192	1.0	*utmost, sheer, immense, tremendous, newfound, unparalleled, ...*
IMDB	NNSE	*neg*	217	1.0	*debris, trash, garbage, lint, rubbish, sludge, dust, dirt, manure, ...*
IMDB	NMF300	*pos*	100	1.0	*imaginative, vivid, lyrical, poetic, realistic, imagery, subtle, ...*
IMDB	NMF300	*pos*	131	0.76	*shakira, lauper, mcentire, yearwood, parton, estefan, streisand, ...*
BoolQ	SPINE	*false*	575	1.0	*leaked, confidential, libby, fbi, classified, memo, leak, intelligence, ...*
BoolQ	SPINE	*false*	841	0.79	*astronaut, soyuz, spacecraft, iss, nasa, astronauts, shuttle, mir, ...*
BoolQ	SPOWV	*true*	758	1.0	*cyclone, katrina, hurricane, disaster, ike, flooded, shear, dolly, ...*
BoolQ	SPOWV	*true*	173	0.83	*tong, lumpur, myanmar, singaporean, kuala, chung, penang, ...*

TABLE 5 – Exemples de paramètres de prédiction appris par les modèles élémentaires : D = Jeu de Données ; M = Modèle de plongements ; C = Classe correspondant au paramètre ; I = Indice de la dimension correspondant au paramètre dans le modèle de plongements ; P = Poids du paramètre, divisé par la valeur du plus grand paramètre (pour cette classe).

IMDB : Il s'agit de l'un des jeux de données pour lequel les performances des modèles élémentaires entraînés sont les plus élevées. Sans trop de surprises, une partie significative des dimensions les plus actives pour les classes "positive" et "négative" semblent correspondre à des champs lexicaux contenant des marqueurs de sentiment appropriés, pour la plupart des modèles. Cependant, pour le modèle NMF300 en particulier, nous avons remarqué plusieurs dimensions associées à un grand nombre de noms de famille et de prénoms (par exemple, 4ème ligne de la Table 5) et qui apparaissent comme prédicteurs forts de la classe "positive". Pour analyser ce biais a priori peu pertinent dans l'absolu pour une tâche d'analyse de sentiment, nous avons utilisé le module de reconnaissance d'entités nommées de la bibliothèque spaCy pour compter les entités nommées de type "personne" dans les critiques du jeu de données, et nous avons constaté une corrélation linéaire faible (coefficient de Pearson $r = 0.124$) entre ces comptages et les classes des instances. Une analyse plus poussée serait nécessaire pour prouver si un modèle non-linéaire exploite en effet cet aspect, ou un aspect plus spécifique encore : il semble en effet que plusieurs dimensions créées par NMF300 sont caractérisées par des noms d'artistes célèbres. On peut noter que 80.68% des critiques du jeu de données contiennent au moins une entité nommée de ce type, ce qui est cohérent avec la pondération élevée du paramètre correspondant à cette dimension.

BoolQ : Sur ce jeu de données, les poids les plus forts portent essentiellement sur des thèmes particuliers : pour les réponses "fausses" ce sont des questions très débattues (de régime, de lois, etc.),

5. https://github.com/TomBourgeade/InterpEmbsForBiasDetection/tree/main/experiments_results

souvent sujette au conspirationnisme (services de renseignements, conquête/exploration spatiale, etc.), ou marquées émotionnellement (avec des adjectifs *dignified* ou des adverbes *dramatically*). Pour les réponses "vraies", ce sont des dimensions plus liées à la science, l'histoire, la géographie, ou la politique, ou des valeurs numériques. Tout ceci peut indiquer un léger biais dans la collecte qui repose sur les requêtes faites par les utilisateurs sur un moteur de recherche, qui est peut-être exploité par les modèles sans qu'ils aient besoin d'analyser la réponse. Néanmoins, comme il s'agit d'une tâche à deux entrées (question et passage), nous pouvons également observer dans quelle partie du vecteur de composition $< u, v, |u - v|, u * v >$ (voir Section 5.2) se trouve les paramètres correspondants les plus importants : pour la plupart des modèles (à l'exception notable de `NMF300` et `Word2Sense`), nous observons que ceux-ci se trouvent dans la partie produit terme-à-terme de la composition, ce qui indique que les modèles élémentaires se base principalement sur des interactions entre la question et le passage dans l'entrée, ce qui est attendu vis-à-vis de la tâche. Les paramètres importants restants sont en revanche le plus souvent situé dans la partie de la composition qui correspond à la question seule, ce qui pourrait indiquer la présence de questions plus ou moins "rhétoriquement" biaisées.

Emergent : On trouve ici également des thématiques cohérentes qui correspondent à des sujets peu controversés. Par exemple une dimension reliée aux animaux est fortement corrélée avec le label "pour" : une revue rapide des articles de journaux du jeu de données confirme que ceux-ci sont pratiquement toujours positifs sur ces sujets. Ce genre de biais semble inhérent à la façon dont le jeu de données est construit à partir de titres de journaux.

Sarcasm et UR-FUNNY : Les données `Sarcasm` révèlent quelques sujets populaires en regardant les résultats avec `NMF300`, les dimensions positives étant associées aux artistes musicaux, avec par exemple les mots les plus actifs suivants *"burnin, dreamin, rmx, blowin, movin"*, ou bien *"lil, ludacris, rapper, dogg, snoop"*, les dimensions négatives étant liées à des sujets médicaux, par exemple la dimension dont les mots actifs sont *"neurology, ophthalmology, oncology"*, ou légaux (*"plaintiffs, plaintiff, court, appeals"*, ou de manière général techniques, ce qui peut indiquer un manque de diversité. On observe des éléments similaires sur les données `UR-FUNNY` notamment avec le modèle `NMF300`, mais en se focalisant plus sur la chute du sketch. Le modèle `NNSE` montre plus de variétés, et des poids mieux répartis, se focalisant plus sur les traits composés des deux représentations en entrée. Les prédicteurs négatifs incluent toujours des dimensions à l'évidence techniques.

PDTB : La prédiction de relations implicites est intéressante car elle mixe des relations sémantiques et pragmatiques difficiles. Tous les modèles ne prédisent que 4-5 des relations, les plus fréquentes : *Cause*, *Contrast*, *Conjunction*, *Restatement* et *Instantiation*. L'analyse des instances semble montrer quelques particularités liés au type de textes (journalistiques). Par exemple, si l'on regarde plus en détail la relation d'Instantiation, elle semble dépendre beaucoup d'une dimension où les termes les plus actifs sont *"educator, historian, lecturer, researcher, scientist, essayist, journalist, curator, critic, playwright"* dans le second argument de la relation. En regardant les exemples de l'instance d'entraînement, nous avons observé que les seuls mots de cette liste qui apparaissent avec cette relation sont *"critic, journalist, scientist"*, dans une vingtaine d'instances. Cela semble indiquer que ce sont principalement des citations qui illustrent un point mentionné dans le premier argument de la relation. Cela est confirmé quand on analyse les autres signaux de citations, qui constituent un tiers de toutes les instances d'Instantiation, ce qui semble très typique de textes journalistiques mais sans doute peu représentatif au-delà. De la même façon, certaines dimensions importantes pour la prédiction des autres classes de relations semblent assez spécifique pour justifier une étude détaillée des exemples traités.

SNLI : Ce jeu de données semble être un cas particulier, avec de nombreuses dimensions différentes, interprétables mais sans liens évidents entre elles (pour tous les modèles), ce qui pourrait s'expliquer en partie par la taille conséquente et donc la variété des données. SNLI a des biais connus (voir section 2), qui sont en partie associés à la syntaxe (négations, phrases prépositionnelles supplémentaires). Ceux-ci sont bien sûr plus difficiles à découvrir avec les plongements utilisés ici, qui sont principalement lexicaux par nature.

Nous constatons que cette méthode d'analyse semble relativement intéressante pour rapidement détecter des corrélations lexicales faciles à apprendre, mais ne suffit pas seule à confirmer leurs natures exactes, leurs magnitudes, ou si elles ont effectivement pour cause la présence de biais erronés dans les données. La création de protocoles de diagnostic plus poussés, guidés par les corrélations relevées ici a priori, serait nécessaire pour confirmer ces aspects, mais ceux-ci pourraient requérir de lourds moyens humains, en particulier s'ils nécessitent des analyses qualitatives d'un grand nombre d'instances.

6 Conclusion et perspectives

Nous avons montré ici comment une méthode simple peut être utilisée pour identifier des biais non voulus dans des données de TAL, en exploitant des plongements lexicaux interprétables. Cela peut permettre de repérer des problèmes avant de mettre en jeu des modèles plus complexes, ou en amont de tâches différentes. Les évaluations intrinsèques et extrinsèques montrent que des plongements interprétables plus récents ont de meilleures performances sur certaines tâches, mais sans pour autant que l'interprétabilité de leurs dimensions soient meilleures.

Parmi les améliorations envisagées, la principale serait de pouvoir déterminer comment évaluer par l'humain les explications produite par une approche, en s'inspirant par exemple des évaluations d'explications de Strout *et al.* (2019).

Une limite de l'approche présentée est de n'avoir accès qu'à des phénomènes lexicaux, ce qui empêcherait de repérer des biais plus structurels (syntaxiques ou discursifs par exemple). Une avenue prometteuse serait de garder des modèles permettant d'encoder des informations de ce type, comme le modèle Transformer BERT (Devlin *et al.*, 2019) et ses descendants, en les entraînant d'une façon interprétable.

Enfin il serait utile de combiner l'approche avec les méthodes qui se focalisent sur l'interprétation d'instances : en plus de repérer des biais au niveau global d'un jeu de données, pouvoir identifier les instances spécifiques qui en sont responsables permettrait de concevoir des protocoles de diagnostic plus poussés, ciblant spécifiquement une ou plusieurs parties problématiques dans les données, afin d'en déterminer les causes et potentiellement les réparer. Une méthode comme la Layer-wise Relevance Propagation (Montavon *et al.*, 2019) par exemple peut diagnostiquer un modèle "suspect" sur les instances identifiées comme source potentielle de biais, permettant de corriger le problème dans les données, ou dans le modèle directement.

Références

AKIBA T., SANO S., YANASE T., OHTA T. & KOYAMA M. (2019). Optuna : A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

ARORA S., LIANG Y. & MA T. (2017). A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

BERGSTRA J., BARDENET R., BENGIO Y. & KÉGL B. (2011). Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, p. 2546–2554.

BERGSTRA J., YAMINS D. & COX D. D. (2013). Making a science of model search : Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, p. I–115–I–123.

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. DOI : 10.1162/tacl_a_00051.

BOWMAN S. R., ANGELI G., POTTS C. & MANNING C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 632–642, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1075.

CHANG J., GERRISH S., WANG C., BOYD-GRABER J. L. & BLEI D. M. (2009). Reading Tea Leaves : How Humans Interpret Topic Models. In Y. BENGIO, D. SCHUURMANS, J. D. LAFFERTY, C. K. I. WILLIAMS & A. CULOTTA, Éds., *Advances in Neural Information Processing Systems 22*, p. 288–296. Curran Associates, Inc.

CLARK C., LEE K., CHANG M.-W., KWIATKOWSKI T., COLLINS M. & TOUTANOVA K. (2019). BoolQ : Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 2924–2936, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1300.

CONNEAU A., KIELA D., SCHWENK H., BARRAULT L. & BORDES A. (2017). Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 670–680, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/D17-1070.

DAHIYA Y., TALUKDAR P. & OTHERS (2016). Discovering response-eliciting factors in social question answering : A reddit inspired study. In *Tenth International AAAI Conference on Web and Social Media*.

DAI Z. & HUANG R. (2019). A regularization approach for incorporating event knowledge and coreference relations into neural discourse parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 2976–2987, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1295.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference*

of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers), p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DUFTER P. & SCHÜTZE H. (2019). Analytical Methods for Interpretable Ultradense Word Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 1185–1191, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1111.

FARUQUI M., TSVETKOV Y., YOGATAMA D., DYER C. & SMITH N. A. (2015). Sparse Overcomplete Word Vector Representations. 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)*, p. 1491–1500, Beijing, China : Association for Computational Linguistics. DOI : 10.3115/v1/P15-1144.

FERREIRA W. & VLACHOS A. (2016). Emergent : a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1163–1168, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1138.

FYSHE A., TALUKDAR P. P., MURPHY B. & MITCHELL T. M. (2014). Interpretable Semantic Vectors from a Joint Model of Brain- and Text- Based Meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 489–499, Baltimore, Maryland : Association for Computational Linguistics. DOI : 10.3115/v1/P14-1046.

GURURANGAN S., SWAYAMDIPTA S., LEVY O., SCHWARTZ R., BOWMAN S. & SMITH N. A. (2018). Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 107–112, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-2017.

HASAN M. K., RAHMAN W., BAGHER ZADEH A., ZHONG J., TANVEER M. I., MORENCY L.-P. & HOQUE M. E. (2019). UR-FUNNY : A Multimodal Language Dataset for Understanding Humor. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 2046–2056, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1211.

HE H., ZHA S. & WANG H. (2019). Unlearn Dataset Bias in Natural Language Inference by Fitting the Residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, p. 132–142, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-6115.

HOSSAIN M. M., KOVATCHEV V., DUTTA P., KAO T., WEI E. & BLANCO E. (2020). An Analysis of Natural Language Inference Benchmarks through the Lens of Negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 9106–9118, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.732.

HOYER P. (2002). Non-negative sparse coding. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, p. 557–565. DOI : 10.1109/NNSP.2002.1030067.

HURTADO BODELL M., ARVIDSSON M. & MAGNUSSON M. (2019). Interpretable Word Embeddings via Informative Priors. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

Processing (EMNLP-IJCNLP), p. 6324–6330, Hong Kong, China : Association for Computational Linguistics.

KASSNER N. & SCHÜTZE H. (2020). Negated and Misprimed Probes for Pretrained Language Models : Birds Can Talk, But Cannot Fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7811–7818, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.698.

KOBER T., WEEDS J., WILKIE J., REFFIN J. & WEIR D. (2017). One Representation per Word - Does it make Sense for Composition ? In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, p. 79–90, Valencia, Spain : Association for Computational Linguistics. DOI : 10.18653/v1/W17-1910.

LEE D. D. & SEUNG H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**(6755), 788–791. DOI : 10.1038/44565.

LEE D. D. & SEUNG H. S. (2001). Algorithms for Non-negative Matrix Factorization. In T. K. LEEN, T. G. DIETTERICH & V. TRESP, Éds., *Advances in Neural Information Processing Systems 13*, p. 556–562. MIT Press.

LI J., CHEN X., HOVY E. & JURAFSKY D. (2016). Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 681–691, San Diego, California : Association for Computational Linguistics.

LIU X., HE P., CHEN W. & GAO J. (2019). Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4487–4496, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1441.

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 : Human Language Technologies*, p. 142–150, Portland, Oregon, USA : Association for Computational Linguistics.

MCCOY T., PAVLICK E. & LINZEN T. (2019). Right for the Wrong Reasons : Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3428–3448, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1334.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. BURGES, L. BOTTOU, M. WELLING, Z. GHAHRAMANI & K. Q. WEINBERGER, Éds., *Advances in Neural Information Processing Systems 26*, p. 3111–3119. Curran Associates, Inc.

MONTAVON G., BINDER A., LAPUSCHKIN S., SAMEK W. & MÜLLER K.-R. (2019). Layer-Wise Relevance Propagation : An Overview. In W. SAMEK, G. MONTAVON, A. VEDALDI, L. K. HANSEN & K.-R. MÜLLER, Éds., *Explainable AI : Interpreting, Explaining and Visualizing Deep Learning*, p. 193–209. Cham : Springer International Publishing. DOI : 10.1007/978-3-030-28954-6_10.

MURPHY B., TALUKDAR P. & MITCHELL T. (2012). Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proceedings of COLING 2012*, p. 1933–1950, Mumbai, India : The COLING 2012 Organizing Committee.

NAIK A., RAVICHANDER A., SADEH N., ROSE C. & NEUBIG G. (2018). Stress Test Evaluation for Natural Language Inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 2340–2353, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

ORABY S., HARRISON V., REED L., HERNANDEZ E., RILOFF E. & WALKER M. (2016). Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 31–41, Los Angeles : Association for Computational Linguistics. DOI : 10.18653/v1/W16-3604.

PANIGRAHI A., SIMHADRI H. V. & BHATTACHARYYA C. (2019). Word2Sense : Sparse Interpretable Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 5692–5705, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1570.

PARK S., BAK J. & OH A. (2017). Rotated Word Vector Representations and their Interpretability. In M. PALMER, R. HWA & S. RIEDEL, Éds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, p. 401–411 : Association for Computational Linguistics. DOI : 10.18653/v1/d17-1041.

PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1162.

POLIAK A., NARADOWSKY J., HALDAR A., RUDINGER R. & VAN DURME B. (2018). Hypothesis Only Baselines in Natural Language Inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, p. 180–191, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/S18-2023.

PRASAD R., DINESH N., LEE A., MILTSAKAKI E., ROBALDO L., JOSHI A. & WEBBER B. (2008). The Penn Discourse TreeBank 2.0. In *Proceedings of LREC*.

RIBEIRO M. T., SINGH S. & GUESTRIN C. (2016). "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, p. 1135–1144, San Francisco, California, USA : ACM Press. DOI : 10.1145/2939672.2939778.

SOCHER R., PERELYGIN A., WU J., CHUANG J., MANNING C. D., NG A. & POTTS C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 1631–1642, Seattle, Washington, USA : Association for Computational Linguistics.

STROUT J., ZHANG Y. & MOONEY R. (2019). Do Human Rationales Improve Machine Explanations ? In *Proceedings of the 2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 56–62, Florence, Italy : Association for Computational Linguistics.

SUBRAMANIAN A., PRUTHI D., JHAMTANI H., BERG-KIRKPATRICK T. & HOVY E. H. (2018). SPINE : SParse Interpretable Neural Embeddings. In S. A. MCILRAITH & K. Q. WEINBERGER, Éds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, p. 4921–4928 : AAAI Press.

TORRALBA A. & EFROS A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011*, p. 1521–1528, Colorado Springs, CO, USA : IEEE. DOI : 10.1109/CVPR.2011.5995347.

TRIFONOV V., GANEA O.-E., POTAPENKO A. & HOFMANN T. (2018). Learning and Evaluating Sparse Interpretable Sentence Embeddings. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 200–210, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5422.

WILLIAMS A., NANGIA N. & BOWMAN S. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1112–1122, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1101.

YANG Z., DAI Z., YANG Y., CARBONELL J., SALAKHUTDINOV R. R. & LE Q. V. (2019). XLNet : Generalized Autoregressive Pretraining for Language Understanding. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ BUC, E. FOX & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 32*, p. 5753–5763. Curran Associates, Inc.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Transport optimal pour le changement sémantique à partir de plongements contextualisés

Syrielle Montariol[1]* Alexandre Allauzen[2]
(1) INRIA Paris, France
(2) PSL, Dauphine-Université et ESPCI, Paris, France
syrielle.montariol@inria.fr, alexandre.allauzen@espci.psl.eu

RÉSUMÉ

Plusieurs méthodes de détection des changements sémantiques utilisant des plongements lexicaux contextualisés sont apparues récemment. Elles permettent une analyse fine du changement d'usage des mots, en agrégeant les plongements contextualisés en clusters qui reflètent les différents usages d'un mot. Nous proposons une nouvelle méthode basée sur le transport optimal. Nous l'évaluons sur plusieurs corpus annotés, montrant un gain de précision par rapport aux autres méthodes utilisant des plongements contextualisés, et l'illustrons sur un corpus d'articles de journaux.

ABSTRACT

Optimal Transport for Semantic Change Detection using Contextualised Embeddings

Several methods for semantic change detection with contextualised embeddings emerged recently. They allow a fine-grained analysis of word usage change by aggregating embeddings into clusters that reflect different usages of a word. We propose a novel method based on optimal transport. We evaluate it on several annotated corpora, showing a gain in accuracy compared to other methods using contextualised embeddings, and illustrate it on a corpus of newspaper articles.

MOTS-CLÉS : Changement sémantique, Transport optimal, Plongements contextualisés.

KEYWORDS: Semantic change, Optimal transport, Contextualised embeddings.

1 Introduction

La *diachronie* désigne l'évolution du langage à travers le temps. L'un des aspects de la diachronie est l'évolution de la signification des mots. Détecter et comprendre ces changements sémantiques est utile, par exemple, en sociolinguistique et en linguistique historique. Ce domaine a rapidement évolué avec l'essor de la sémantique distributionnelle ; les modèles de plongements lexicaux diachroniques ont connu une vague d'intérêt ces dernières années (Tahmasebi *et al.*, 2018). Ils sont utilisés pour des tâches d'analyse de flux de texte, telles que la détection d'événements (Kutuzov *et al.*, 2017) ou la surveillance des changements de discours lors de crises (Stewart *et al.*, 2017).

Suite à l'émergence des plongements lexicaux (e.g. Word2Vec, (Mikolov *et al.*, 2013)), une approche classique pour le changement sémantique implique d'apprendre des plongement pour chaque strate temporelles d'un corpus et de les rendre comparables en alignant les espaces vectoriels (Hamilton *et al.*, 2016). Cette méthode s'est révélée efficace sur un ensemble de dérives sémantiques synthé-

* Travaux effectués au sein du laboratoire LISN-CNRS en partenariat avec l'Université Paris-Saclay.

tiques (Shoemark *et al.*, 2019) et a été largement utilisée dans la littérature (Dubossarsky *et al.*, 2019; Schlechtweg *et al.*, 2020). Une autre approche compare les voisins d'un mot dans son espaces de représentation à différentes périodes (Yin *et al.*, 2018; Gonen *et al.*, 2020). Dans toutes ces méthodes, chaque mot n'a qu'une seule représentation dans une tranche de temps, ce qui limite la sensibilité et l'interprétabilité de ces techniques. Les plongements contextualisés issus de modèles de langues tels que BERT (Devlin *et al.*, 2019) permettent à chaque occurrence de mot d'avoir une représentation vectorielle qui lui est propre. Des travaux récents montrent que de tels plongements peuvent être utilisés pour la détection des changements sémantiques, en agrégeant les informations de l'ensemble des plongements d'un mot selon différentes méthodes (Martinc *et al.*, 2020b; Giulianelli *et al.*, 2020).

Dans cet article, nous résumons ces méthodes et proposons une nouvelle approche basée sur le transport optimal. Bien qu'ancienne (Monge, 1781), la théorie du transport optimal a connu des avancées importantes au XXe siècle, notamment avec Brenier (1987) qui lie le problème avec la théorie des probabilités. Avec l'essor de l'apprentissage automatique, les applications du transport optimal se sont élargies. Dans le cadre du TAL, il est appliqué au transport de mots ou d'ensembles de mots pour diverses tâches : alignement d'espaces de représentation (Alvarez-Melis & Jaakkola, 2018; Alaux *et al.*, 2019), classification de documents (Kusner *et al.*, 2015), mais aussi apprentissage de représentation et *topic modelling* (Xu *et al.*, 2018). Dans le cadre de l'évolution langagière, Huang & Paul (2019) utilisent la distance de Wasserstein pour mesurer le changement sémantique entre plusieurs corpus. La détection de variation sémantique au niveau lexical est un nouveau cadre d'application prometteur. Nous évaluons notre méthode et la comparons avec celles de la littérature sur plusieurs corpus annotés, puis l'appliquons à un corpus d'articles de journaux pour l'illustrer.

2 Méthodologie

Nous utilisons un modèle BERT pré-entraîné pour extraire des plongements contextualisés [1] d'un corpus divisé en strates temporelles, pour une liste de mots-cibles. Ces plongements peuvent être comparés entre deux strates temporelles pour mesurer le degré de changement sémantique du mot. Une première méthode de comparaison, le "moyennage", consiste à moyenner tous les plongements contextualisés d'un mot apparaissant à une période donnée (Martinc *et al.*, 2020b). On obtient une unique représentation vectorielle du mot pour chaque période ; ces vecteurs peuvent être comparés à l'aide de la distance cosinus (DC). [2] Dans la section qui suit, nous présentons une seconde méthode basée sur le *clustering* des plongements, sur laquelle s'appuie notre méthode de transport optimal.

2.1 Méthode de clustering

Nous effectuons un clustering sur tous les plongements contextualisés d'un mot, et considérons chaque cluster comme un usage du mot. Nous en déduisons la distribution des usages à chaque période. Un algorithme communément utilisé pour cette tâche est la propagation par affinité (Martinc *et al.*, 2020a), un clustering itératif qui déduit automatiquement le nombre de clusters pendant l'entraînement (Frey & Dueck, 2007). Néanmoins, les clusters sont généralement nombreux et de tailles très inégales.

1. Nous obtenons des plongements contextualisés en additionnant les quatre dernières couches de sortie des encodeurs de BERT. Le plongement d'un mot est calculé à partir de la moyenne des plongements de ses *wordpieces*.

2. Par abus de langage, nous définissons ici la distance cosinus comme le complément de la similarité cosinus dans l'espace des réels positifs (1 - similarité cosinus).

Pour surmonter cette limitation — diminuer le nombre de clusters a posteriori, afin de se concentrer sur les usages "principaux" des mots tout en limitant la perte d'information — nous utilisons la méthode de *filtrage* de Montariol *et al.* (2021). Un cluster est représenté par la moyenne de tous les plongements qu'il contient. Nous fusionnons chaque cluster avec le cluster le plus proche selon la distance cosinus entre leurs moyennes. Si le plus proche se trouve à une distance supérieure à un seuil [3] et que le cluster comporte moins de 10 éléments, [4] alors il est supprimé. Cette procédure est appliquée de manière récursive jusqu'à ce que la distance minimale entre deux clusters soit supérieure au seuil, ou qu'il ne reste que 2 clusters.

Pour un mot donné, le clustering est effectué sur les plongements issus de toutes les strates temporelles conjointement, afin de déduire une distribution de clusters unique pour toutes les périodes. Les distributions sont normalisées par la fréquence des mots dans leur strate, et comparées en utilisant la divergence de Jensen-Shannon (DJS) (Lin, 2006).

2.2 Méthode du transport optimal

La méthode du moyennage conserve la dimension originale des plongements de BERT (768) et permet une comparaison précise du contexte moyen d'un mot entre strates temporelles ; mais l'information sur la diversité du contexte intra-période est perdue. À l'inverse, la méthode de clustering capture la variabilité du contexte d'un mot en décomposant ses représentations en une distribution de faible dimension ; cependant, l'information sémantique apprise par le modèle et enregistrée dans les plongements est perdue. Pour conserver les deux types d'informations lors de la comparaison de l'usage d'un mot entre deux périodes, nous nous appuyons sur le cadre du transport optimal.

Formulation. L'ensemble des plongements contextualisés d'un mot sont regroupés ; soit avec un clustering unique comme dans la section précédente, soit avec un clustering différent pour les plongements de chaque période. Ensuite, nous calculons la moyenne de tous les plongements d'une période à l'intérieur d'un cluster. Dans une situation avec K clusters et T périodes, on obtient une matrice de plongements (de taille $T \times K \times 768$) et une distribution des clusters ($T \times K$) pour chaque mot. Nous résumons ainsi toutes les informations du nuage de plongements contextualisés à chaque période en K points dans un espace de dimension 768, les *centroïdes*, pondérés par le nombre de plongements dans le cluster associé. Nous souhaitons comparer ces centroïdes entre les périodes.

Cette configuration peut être formulée de la manière suivante. On note $\mu^{(1)}, \mu^{(2)} \in \mathbb{R}^{K\times 768}$ les ensembles de K centroïdes dans les deux périodes, et $c^{(1)}, c^{(2)} \in \Delta^{K-1}$ les distributions marginales des clusters telles que $c_i^{(t)} = p(C = i|\mathcal{T} = t, w)$ est la distribution du cluster i des plongements du mot w pour la période t. Δ^{K-1} est le simplexe $K-1$ standard : $c^{(1)}$ et $c^{(2)}$ sont des vecteurs positifs de dimension K et se somment à 1. Ils représentent les poids de chaque centroïde dans les espaces source et cible ($\mu^{(1)}$ et $\mu^{(2)}$). Nous quantifions l'effort de déplacement d'une unité de masse d'un centroïde de $\mu^{(1)}$ à un centroïde de $\mu^{(2)}$ avec une fonction de coût, ici la distance cosinus. Nous résolvons alors le problème en recherchant l'effort minimal requis pour transformer la distribution de masse de $c^{(1)}$ sur $\mu^{(1)}$ en celle de $c^{(2)}$ sur $\mu^{(2)}$.

3. Le seuil est égal à $moy_{dc} - 2 \times ect_{dc}$, où moy_{dc} est la moyenne des distances cosinus entre les clusters et ect_{dc} est l'écart-type. Les valeurs limitées par ce seuil représentent environ 95% d'une distribution normale.

4. Le nombre de 10 est choisi à partir de la procédure d'annotation la tâche SemEval2020–1 (Schlechtweg *et al.*, 2020), où chaque sens doit être annoté au moins 5 fois dans une période afin d'être validé. Nous expérimentons sur des corpus divisés en 2 périodes, d'où le choix de 10 instances.

Distance de Wasserstein (DW). Le transport optimal, également appelé problème de Monge-Kantorovitch, a pour but de résoudre ce problème d'optimisation. Il peut être formulé et résolu avec la programmation linéaire. Ici, nous donnons un bref aperçu du cadre de ce problème ; pour plus de détails, nous renvoyons le lecteur à des articles tels que (Villani, 2004; Solomon, 2018). La DW est positive, symétrique et satisfait l'inégalité triangulaire : autant de propriétés qui en font une distance. Pour notre tâche, elle peut être calculée de la manière suivante :

$$W(c^{(1)}, c^{(2)}) = \min_{\gamma} \sum_{i,j} \gamma_{ij} \cos(\mu_i^{(1)}, \mu_j^{(2)}) \text{ avec } \gamma \mathbf{1} = c^{(1)};\ \gamma^\intercal \mathbf{1} = c^{(2)};\ \gamma \geq 0 \qquad (1)$$

En d'autres termes, nous voulons minimiser le travail total ($\min_\gamma$) pour aller de $c^{(1)}$ à $c^{(2)}$ à l'aide de la distance cosinus ($\cos$), étant donné que la masse transportée est positive ($\gamma \geq 0$). La résolution de cette équation conduit à un plan de transport γ. Elle peut être vue comme une fonction de masse de probabilité sur $K \times K$ dont les marginales sont $c^{(1)}$ et $c^{(2)}$, et qui quantifie la proportion de masse $c_i^{(1)}$ de $\mu_i^{(1)}$ devant être transférée vers $\mu_j^{(2)}$ afin d'obtenir une masse $c_j^{(2)}$ de la manière la plus efficace. La DW représente la somme de tout le travail nécessaire pour résoudre le problème.

Notons que ce problème est complètement différent de la configuration de la section précédente résolue avec la divergence de Jensen-Shannon ; au lieu de comparer deux distributions, nous comparons deux ensembles pondérés de centroïdes. C'est pourquoi nous n'avons pas besoin d'avoir les mêmes clusters pour toutes les strates temporelles ; deux clusterings indépendants, un par période, pourraient permettre un meilleur ajustement pour chaque ensemble de points sans nuire au calcul de la distance.

3 Évaluation

Extraction des plongements contextualisés. Afin de résumer l'information efficacement, au lieu de garder en mémoire autant de vecteurs que d'occurrences d'un mot, nous utilisons une méthode de regroupement-moyenne (Montariol *et al.*, 2021) en ne stockant qu'un nombre limité de plongements (ici 200) pour chaque strate. À chaque nouvelle occurrence du mot dans la strate, son plongement e_{new} est additionné au vecteur e_m qui est lui est le plus similaire dans la liste de 200 plongements [5]. Le nombre d'éléments ajoutés dans e_m est incrémenté (compteur $c_m \leftarrow c_m + 1$) pour normaliser chaque élément de la liste de plongements à la fin de l'extraction.

Données annotées et modèles. Nous utilisons six jeux de données annotés pour l'évaluation : un jeu en anglais appelé "GEMS", quatre issus d'une tâche d'évaluation SemEval, et "DURel", un jeu en allemand. GEMS (Gulordava & Baroni, 2011) est nommé ainsi après le workshop GEMS où l'article associé a été publié. 5 annotateurs ont évalué le degré de changement sémantique de 100 mots anglais entre les années 1960 et 1990, sans observer les mots en contexte. Afin d'étudier l'évolution sémantique de ces mots, nous générons des plongements contextualisés à partir de textes des décennies 1960 et 1990 du Corpus of Historical American English (COHA) [6] (2,8M et 3,3M de mots respectivement). La tâche SemEval 2020 – 1 : Détection non supervisée de changement lexico-sémantique (Schlechtweg *et al.*, 2020) propose des données annotées en utilisant une nouvelle

5. En utilisant la distance cosinus : $e_m = \arg\min_{e_i \in L} \cos(e_i, e_{\text{new}})$.
6. https://www.english-corpora.org/coha/

Méthode	Mesure	GEMS	SemEval				DURel	Moy
			Anglais	Allemand	Suédois	Latin		
base	DW	0,312	0,386	0,416	0,252	0,283	0,526	0,363
clustering	DW	0,369	**0,456**	0,421	0,264	0,397	0,484	0,399
2× clustering	DW	0,380	0,412	0,457	0,190	0,426	0,530	0,399
clustering + filtrage	DW	0,352	0,437	0,561	0,321	0,488	0,686	0,474
clustering	DJS	**0,394**	0,371	0,498	0,012	0,346	0,512	0,355
clustering + filtrage	DJS	0,403	0,348	**0,583**	0,018	0,408	0,712	0,412
moyennage	DC	0,349	0,315	0,565	0,212	**0,496**	0,656	0,432
SGNS + PO	DC	0,347	0,321	0,712	**0,631**	0,372	**0,814**	**0,533**

TABLE 1 – Corrélation de Spearman entre les classements de chaque système et le classement issu de l'annotation, pour chaque corpus de test. Les valeurs grisées indiquent une corrélation non significative (p-valeur > 0,05). [10]

approche : les annotateurs décident si une paire de phrases de différentes périodes portent la même signification du mot-cible (Schlechtweg & Schulte im Walde, 2020). Les corpus, en quatre langues — anglais (13,4M de mots), allemand (142M), suédois (182M) et latin (11,2M) — sont divisés en deux périodes et les phrases sont mélangées et lemmatisées. Enfin, le jeu de données DURel [7] (Schlechtweg *et al.*, 2018) est composé de 22 mots allemands, classés selon leur degré de changement sémantique entre deux périodes par cinq annotateurs selon la même méthode que pour SemEval. Nous générons des plongements en utilisant le corpus allemand DTA, également lemmatisé (25M de mots pour 1750–1799 et 38M pour 1850–1899). L'adaptation au domaine (tâche *masked language model*) est effectuée sur chaque corpus pour 5 itérations, en utilisant des modèles BERT pré-entraînés adaptés à chaque langue. [8] Les mots-cibles sont classés en fonction de leur degré de changement sémantique à l'aide des méthodes décrites précédemment. Le classement est comparé avec la vérité terrain à l'aide de la corrélation de Spearman.

Résultats. Nous appliquons différentes méthodes reposant sur le transport optimal pour calculer l'évolution de l'usage d'un mot entre deux périodes de temps (Table 1). [11] Nous pouvons soit faire un clustering unique sur les plongements des deux périodes, soit un clustering différent pour chaque période (noté "2× clustering"). Une autre variation consiste à utiliser le compteur c_m du nombre de plongements ayant été additionnés pour former chacun des 200 vecteurs e_m de la liste, pour pondérer la matrice de coût lors du calcul de la DW sans effectuer de clustering (noté "base"). Nous comparons ces méthodes avec le clustering classique, où la distance est calculée avec la DJS, et avec le moyennage, calculé avec la distance cosinus (DC).

La réalisation de deux clusterings indépendants n'améliore pas les résultats par rapport à un clustering unique, en moyenne. De grands écarts peuvent être observés entre les corpus, dans les deux sens.

7. https://www.ims.uni-stuttgart.de/en/research/resources/experiment-data/durel/

8. Pour l'allemand : bert-base-german-cased (https://deepset.ai/german-bert), pour l'anglais : bert-base-uncased model, pour le latin : bert-base-multilingual-uncased, pour le suédois : bert-base-swedish-uncased (https://github.com/af-ai-center/SweBERT).

10. BERT sur le suédois mène toujours à une corrélation non significative. Nous supposons que ce défaut est dû au modèle utilisé, qui est pré-entraîné sur des textes récents et non lemmatisés ; ils sont donc très éloignés du corpus étudié, composé de textes historiques, lemmatisés et comportant de nombreuses erreurs d'OCR.

11. En utilisant le package POT : https://pythonot.github.io/.

Cependant, comme le nombre de mots-cibles est faible, cela peut s'expliquer par le fait que seuls quelques mots bénéficient ou souffrent du degré de liberté supplémentaire donné par les clusterings indépendants. En effet, sur le plus grand jeu de données de test GEMS (100 mots), les performances entre 1 ou 2 clusterings sont comparables. D'autre part, la comparaison entre les listes complètes de 200 plongements des deux strates sans effectuer de clustering ("base") conduit à une performance moyenne plus faible. L'agrégation apportée par le clustering est donc nécessaire pour limiter la sensibilité au bruit. En outre, le filtrage a un effet positif important sur le clustering, en particulier avec la DW ; en supprimant les clusters minoritaires et extrêmes et en fusionnant les clusters semblables, il vient réduire le bruit et compense le large nombre de clusters inégalement distribués extraits par la propagation par affinité.

Le système SGNS + PO (Schlechtweg *et al.*, 2019) est le seul utilisant des plongements non contextualisés : le modèle Skip-Gram (SGNS) est entraîné sur les deux périodes séparément, et les espaces de représentation sont alignés avec le problème de Procuste orthogonal (PO). La DC est utilisée pour mesurer le changement sémantique. Ce système surpasse largement les autres méthodes en moyenne. Cela peut être lié au fait que les phrases de tous les corpus d'évaluation à l'exception de COHA sont mélangées et lemmatisées. Par conséquent, les modèles BERT ne peuvent exploiter que les phrases au lieu d'une séquence complète de 256 éléments. De plus, SGNS est entraîné intégralement sur les corpus d'évaluation, tandis que les modèles BERT sont pré-entraînés sur du texte brut. Les plongements issus de BERT souffrent dont potentiellement plus de la lemmatisation des corpus.

Parmi les méthodes de plongements contextualisés, le score moyen le plus élevé est obtenu avec clustering + filtrage + DW. Cependant, on trouve de larges disparités selon les jeux de données. Le moyennage surpasse le clustering pour SemEval Latin, tandis que DJS et DW se surpassent alternativement sur les autres corpus. Cette disparité ne semble pas liée à la langue, car des méthodes différentes mènent aux meilleurs scores pour une langue commune (GEMS et SemEval pour l'anglais, DURel et SemEval pour l'allemand). Une cause pourraient être la façon dont chaque méthode distribue les scores de changement sémantique, par rapport à la distribution des scores de la vérité terrain (SemEval latin et allemand et DURel ont des scores de vérité terrain répartis uniformément tandis que SemEval anglais et suédois et GEMS ont une plus forte proportion de scores faibles). En résumé, étant donné que la méthode de transport optimal utilise à la fois les informations du clustering et du moyennage, elle constitue un bon compromis entre ces deux méthodes bien qu'elle ne les surpasse pas systématiquement.

Pour conclure, l'accord inter-annotateurs (moyenne des corrélations de Pearson par paire d'annotateurs) permet de mettre en perspective les performances des méthodes : il est de 0,51 sur le jeu de données GEMS, 0,66 pour DURel et 0,62 pour SemEval (en moyenne sur les 4 jeux de données).

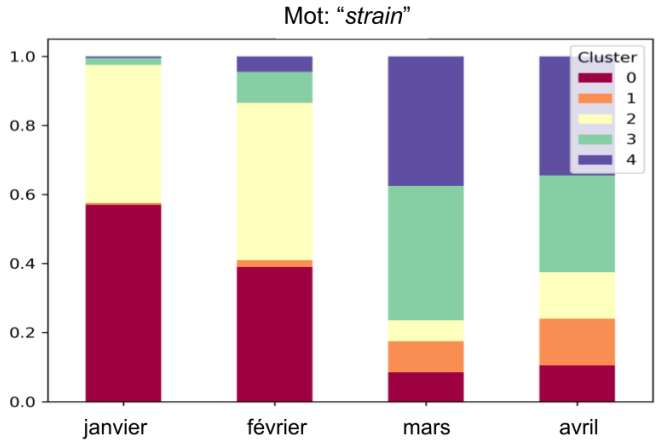
4 Application

Nous montrons un exemple d'exploration diachronique d'un corpus d'articles de journaux en anglais sur le COVID-19. [12] Nous analysons environ 500k articles de janvier à avril 2020, que nous divisons en 4 strates mensuelles de tailles inégales (160M mots en mars, 41M en février, 35M en avril et 10M en janvier). La méthode de transport optimal permet de quantifier l'évolution de l'ensemble des mots du vocabulaire, en extrayant leurs plongements avec la méthode de regroupement-moyenne.

12. https://blog.aylien.com/free-coronavirus-news-dataset/

Nous calculons la DW moyenne entre les mois successifs pour chaque mot du vocabulaire. *Strain* est le 38ème mot avec la DW moyenne la plus élevée, et le 15ème entre février et mars 2020 ; nous nous concentrons dessus pour illustrer les phénomènes d'évolution sémantique pouvant être détectés. *Strain* est un mot polysémique ayant deux sens principaux en anglais apparaissant dans notre corpus : comme la variante d'un virus ou d'une bactérie (terme biologique) et comme "une demande sévère ou excessive sur les ressources ou les capacités de quelqu'un ou de quelque chose" (dictionnaire Oxford). Nous regroupons ses plongements contextuels avec l'algorithme k-means ($k = 5$). Puis, en calculant un score de *tf-idf* sur les unigrammes et les bigrammes des phrases dans lesquelles ce mot apparaît, nous extrayons un ensemble de mots-clés pour chaque cluster – les mots ayant le score de tf-idf le plus élevé – afin d'interpréter les variations de leur distribution (Figure 1).

Les clusters 1, 3 et 4, qui correspondent au deuxième sens du terme (pression sur les systèmes de santé dans le cluster 4, pression financière dans le cluster 3 et pression sur les ressources et les infrastructures dans le cluster 1), voient leur proportion augmenter au fil du temps ; tandis que les clusters 0 et 2, qui correspondent au premier sens du terme (en tant que nouvelle souche de virus), diminuent. Ce comportement souligne l'évolution des préoccupations liées à la pandémie dans les journaux. Ainsi, on observe l'évolution de la répartition des différents sens du mot en terme lexicographique au cours du temps ; mais la méthode permet aussi de révéler les variations d'usage au sein d'une même signification, en fonction des événements de l'actualité.



#	Mots-clés
0	*strain coronavirus, new strain, city wuhan, novel strain, strain virus, chinese city*
1	*strain health, strain resources, stream, network infrastructure, international resources, likely strain*
2	*new strain, acute respiratory, 2019 ncov, respiratory syndrome, severe acute, identified humans*
3	*financial strain, feeling strain, strain coronavirus, economic strain, signs strain, strain said*
4	*ease strain, putting strain, strain health, reduce strain, care system, strain hospitals*

FIGURE 1 – Distributions des clusters par mois et mots-clés principaux pour le mot *strain*.

5 Conclusion

L'évaluation sur données annotées a montré que parmi les méthodes à base de plongements contextualisés, la méthode la plus performante utilise la distance de Wasserstein sur les clusters issus de la propagation par affinité des plongements de BERT. Néanmoins, elle est en moyenne moins performante que la méthode utilisant des plongements non contextualisés (Skip-Gram avec alignement). Malgré ses performances plus faibles, la méthode basée sur le clustering offre une interprétation plus fine que les méthodes basées sur des plongements non contextualisés, car elle tient compte de la diversité des usages et des sens d'un mot ; en particulier, le clustering permet de distinguer les différents usages du mot étudié. C'est pourquoi cette approche peut être utilisée pour détecter l'apparition de nouveaux usages des mots, tracer l'évolution des différents usages, et les interpréter.

Références

ALAUX J., GRAVE E., CUTURI M. & JOULIN A. (2019). Unsupervised hyper-alignment for multilingual word embeddings. In *International Conference on Learning Representations*.

ALVAREZ-MELIS D. & JAAKKOLA T. (2018). Gromov-Wasserstein alignment of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 1881–1890, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1214.

BRENIER Y. (1987). Decomposition polaire et rearrangement monotone des champs de vecteurs. *C. R. Acad. Sci. Paris Ser. I Math.*, **305**, 805–808.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DUBOSSARSKY H., HENGCHEN S., TAHMASEBI N. & SCHLECHTWEG D. (2019). Time-out : Temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 457–470, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1044.

FREY B. J. & DUECK D. (2007). Clustering by passing messages between data points. *Science*, **315**(5814), 972–976.

GIULIANELLI M., DEL TREDICI M. & FERNÁNDEZ R. (2020). Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 3960–3973, Online : Association for Computational Linguistics.

GONEN H., JAWAHAR G., SEDDAH D. & GOLDBERG Y. (2020). Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 538–555, Online : Association for Computational Linguistics.

GULORDAVA K. & BARONI M. (2011). A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, p. 67–71 : Association for Computational Linguistics.

HAMILTON W. L., LESKOVEC J. & JURAFSKY D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 1489–1501. DOI : 10.18653/v1/P16-1141.

HUANG X. & PAUL M. J. (2019). Neural temporality adaptation for document classification : diachronic word embeddings and domain adaptation models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4113–4123, Florence, Italy.

KUSNER M. J., SUN Y., KOLKIN N. I. & WEINBERGER K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, p. 957–966 : JMLR.org.

KUTUZOV A., VELLDAL E. & ØVRELID L. (2017). Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*, p. 31–36, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/W17-2705.

LIN J. (2006). Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, **37**(1), 145–151. DOI : 10.1109/18.61115.

MARTINC M., MONTARIOL S., ZOSA E. & PIVOVAROVA L. (2020a). Capturing evolution in word usage : Just add more clusters ? In *Companion Proceedings of the Web Conference 2020*, WWW '20, p. 343–349, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3366424.3382186.

MARTINC M., NOVAK P. K. & POLLAK S. (2020b). Leveraging contextual embeddings for detecting diachronic semantic shift. *LREC 2020*.

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 26*, p. 3111–3119. Curran Associates, Inc.

MONGE G. (1781). Mémoire sur la théorie des déblais et des remblais. *Imprimerie Royale*.

MONTARIOL S., MARTINC M. & PIVOVAROVA L. (2021). Scalable and interpretable semantic change detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

SCHLECHTWEG D., HÄTTY A., DEL TREDICI M. & SCHULTE IM WALDE S. (2019). A wind of change : Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 732–746, Florence, Italy : Association for Computational Linguistics.

SCHLECHTWEG D., IM WALDE S. S. & ECKMANN S. (2018). Diachronic usage relatedness (durel) : A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 169–174.

SCHLECHTWEG D., MCGILLIVRAY B., HENGCHEN S., DUBOSSARSKY H. & TAHMASEBI N. (2020). Semeval-2020 task 1 : Unsupervised lexical semantic change detection. *SemEval@COLING2020*.

SCHLECHTWEG D. & SCHULTE IM WALDE S. (2020). Simulating lexical semantic change from sense-annotated data. *CoRR*, **abs/2001.03216**.

SHOEMARK P., LIZA F. F., NGUYEN D., HALE S. & MCGILLIVRAY B. (2019). Room to Glo : A systematic comparison of semantic change detection approaches with word embeddings. In *Proceedings of EMNLP-IJCNLP 2019*, p. 66–76, Hong Kong, China : Association for Computational Linguistics.

SOLOMON J. (2018). Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*.

STEWART I., ARENDT D., BELL E. & VOLKOVA S. (2017). Measuring, predicting and visualizing short-term change in word representation and usage in VKontakte social network. In *Eleventh international AAAI conference on web and social media*.

TAHMASEBI N., BORIN L. & JATOWT A. (2018). Survey of computational approaches to diachronic conceptual change. *CoRR*, **1811.06278**.

VILLANI C. (2004). Transport optimal : coup de neuf pour un très vieux problème. In *Images des Mathématiques* : CNRS.

XU H., WANG W., LIU W. & CARIN L. (2018). Distilled wasserstein learning for word embedding and topic modeling. In S. BENGIO, H. WALLACH, H. LAROCHELLE, K. GRAUMAN, N. CESA-BIANCHI & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 31*, p. 1716–1725. Curran Associates, Inc.

YIN Z., SACHIDANANDA V. & PRABHAKAR B. (2018). The global anchor method for quantifying linguistic shifts and domain adaptation. In *Advances in neural information processing systems*, p. 9412–9423.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Vers la production automatique de sous-titres adaptés à l'affichage

François Buet et François Yvon
Université Paris-Saclay, CNRS, LISN,
Campus universitaire bât 508, Rue John von Neumann, F - 91405 Orsay cedex
{francois.buet,francois.yvon}@limsi.fr

RÉSUMÉ

Une façon de réaliser un sous-titrage automatique monolingue est d'associer un système de reconnaissance de parole avec un modèle de traduction de la transcription vers les sous-titres. La tâche de « traduction » est délicate dans la mesure où elle doit opérer une simplification et une compression du texte, respecter des normes liées à l'affichage, tout en composant avec les erreurs issues de la reconnaissance vocale. Une difficulté supplémentaire est la relative rareté des corpus mettant en parallèle transcription automatique et sous-titres sont relativement rares. Nous décrivons ici un nouveau corpus en cours de constitution et nous expérimentons l'utilisation de méthodes de contrôle plus ou moins direct de la longueur des phrases engendrées, afin d'améliorer leur qualité du point de vue linguistique et normatif.

ABSTRACT

Towards automatic adapted monolingual captioning

A possible manner to achieve automatic monolingual closed captioning is to pair an Automatic Speech Recognition (ASR) system with a Machine Translation model turning transcripts into subtitles. The "translation" task is difficult as it must simplify and compress the text, observe norms with respect to display, as well as handle ASR errors. An added difficulty is the relative scarcity of parallel datasets pairing automatic transcripts and subtitles. We describe here the on-going process of corpus collection and we experiment the use of direct or indirect control of the output sentences, in order to improve their quality from a linguistic and normative point of view.

MOTS-CLÉS : Sous-titrage automatique, simplification de textes, traduction automatique.

KEYWORDS: Automatic subtitling, text simplification, machine translation.

1 Introduction

Du fait de l'augmentation générale des sources de contenu audio-visuel, du besoin de leur assurer une diffusion large (en d'autres langues) et des obligations légales concernant l'accessibilité de ces contenus, la production automatique de sous-titres monolingues ou traduits est aujourd'hui un champ d'application très actif du traitement automatique des langues.

La création de sous-titres monolingues nécessite en général d'effectuer une simplification du contenu, de manière à rendre le texte plus abordable pour les lecteurs potentiels. Ceux-ci sont en effet susceptibles de ne pas parfaitement maîtriser la langue écrite ; il peut s'agir par exemple de personnes

ayant une autre langue maternelle, ou bien de personnes sourdes ou malentendantes locutrices de la langue des signes (pour qui l'écrit est assimilable à une langue étrangère) (Daelemans et al., 2004). De plus, les sous-titres doivent satisfaire des contraintes spatiales (les tronçons de phrases doivent rentrer dans la largeur du moniteur, sans trop obstruer le champ de vision) et temporelles (le texte doit être approximativement synchronisé avec les paroles ou l'image, et doit rester affiché suffisamment longtemps pour permettre une lecture confortable à l'écran [1]).

Récemment, les modèles neuronaux ont apporté des avancées significatives dans le domaine de la traduction automatique (Bahdanau et al., 2015; Vaswani et al., 2017), avant d'être adaptés au domaine de la « traduction monolingue », et ont notamment été utilisés pour des tâches de simplification (Zhang et al., 2017; Zhang & Lapata, 2017) et de compression de phrases (Rush et al., 2015; Takase & Okazaki, 2019). Toutefois, ces méthodes demandent de grandes quantités de données parallèles représentant la transformation attendue pour pouvoir être mises en œuvre avec succès. Pour les applications de sous-titrage, les ressources de ce type sont encore relativement lacunaires (Karakanta et al., 2020b).

Nous décrivons ici un nouveau corpus [2] associant des transcriptions automatiques et des sous-titres en français, obtenu à partir du traitement automatique de programmes télévisés contemporains. Ce corpus est utilisé pour mettre en place une chaîne de traitements capable de produire sans intervention humaine le fichier de sous-titres correspondant à une entrée vidéo. Les expériences décrites ici s'intéressent en particulier à l'usage de mécanismes pour contrôler la longueur (Kikuchi et al., 2016; Takase & Okazaki, 2019), et par extension le taux de compression et le débit des phrases engendrées. Nous comparons également différentes stratégies pour mieux contrôler la segmentation du texte en tronçons compatibles avec les normes d'affichage, en fonction du type d'émissions à sous-titrer.

Les questions que nous étudions sont les suivantes :

— Les mécanismes numériques de contrôle de la longueur sont-ils effectifs ?
— Est-il possible d'obtenir un meilleur contrôle en utilisant un marquage symbolique de l'entrée ?
— Est-il utile d'introduire des distinctions entre les émissions qui sont sous-titrées en direct et celles qui sont sous-titrées en post-production ?

Nos expériences mettent en particulier en évidence que les méthodes neuronales utilisées permettent (a) de corriger une partie des erreurs de la reconnaissance vocale ; (b) de calculer des sous-titres respectant globalement les normes d'affichage sans qu'il soit besoin d'expliciter les contraintes de longueur que ces normes imposent.

2 Corpus et métriques

2.1 Corpus

Nous avons à disposition un ensemble de vidéos, assorties de fichiers de sous-titres professionnels, correspondant à des programmes télévisés récemment diffusés en France. Le panel d'émissions qui

1. La *Charte relative à la qualité du sous-titrage à destination des personnes sourdes ou malentendantes* du CSA préconise une fréquence moyenne d'affichage des caractères aux alentours de $12 - 15$ car/s, et un écart maximum de 10 s entre le discours et le sous-titre correspondant (https://www.csa.fr/content/download/20043/334122/version/3/file/Chartesoustitrage122011.pdf, consultée le 14/01/21).

2. La question de la diffusion de cette ressource est délicate : elle appartient au diffuseur pour la partie sous-titre, la propriété des enregistrements étant répartie sur les multiples acteurs de la chaine de production. La question de sa diffusion partielle ou complète n'a pas été décidée et ne nous appartient pas. Le corpus traité continue d'évoluer et de s'accroitre.

nous est fourni a été choisi de manière à représenter diverses catégories (dessin animé, documentaire, fiction, jeu, journal, magazine, politique, vulgarisation).

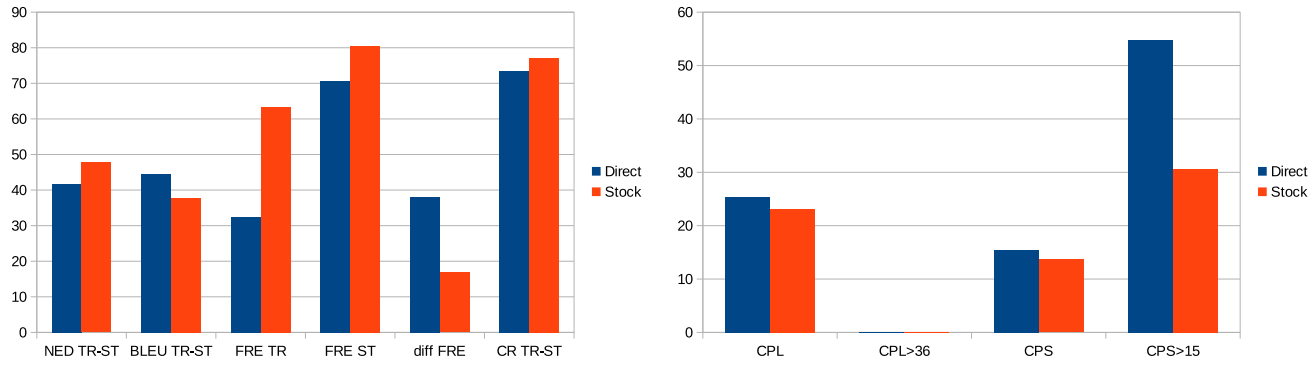
Les instances de programmes collectées, qui arrivent au fur et à mesure des diffusions, sont transcrites automatiquement (mot-pour-mot) en utilisant le système VoxSigma développé conjointement par Vocapia Research et le LIMSI [3]. Ce système délivre des performances à l'état de l'art pour la transcription du français, avec un taux d'erreur proche de 10 % pour de la parole préparée, correspondant par exemple à la transcription de journaux radio-télévisés. Il produit des transcriptions automatiques segmentées automatiquement sur la base d'indices prosodiques et acoustiques (silences, changements de locuteurs, etc) ; les transcriptions sont ponctuées automatiquement, et elles respectent principales les règles typographiques (majuscule en début de phrase, pour les noms propres, etc.). Le texte ainsi obtenu est alors aligné avec celui des sous-titres, afin de pouvoir reconstituer des paires de phrases. Nous avons décidé d'utiliser la segmentation calculée par le système de transcription automatique comme base de l'alignement ; ces segments sont assez longs (environ 40 mots en moyenne), et généralement, correspondent à plusieurs tronçons de sous-titres (voir le Tableau 1). Lors de la réalisation des expériences de la section 3, le corpus contenait environ 411 000 paires de phrases, soit environ 17 millions de mots transcrits, et près de 1600 heures de vidéo. Toutefois, après un filtrage selon la qualité de l'alignement, seulement 265 000 paires ont été utilisées pour l'apprentissage des modèles.

Une distinction notable peut être faite entre les sous-titres provenant d'émissions diffusées en *direct*, et ceux provenant d'émissions de *stock*. Dans le premier cas, les sous-titres sont produits pendant la diffusion, alors que dans le second cas, les sous-titres sont préparés en amont de la diffusion. Ces deux classes sont équitablement réparties dans le corpus (54 % direct et 46 % stock). La figure 1 met en lumière leurs différences selon plusieurs métriques (dont certaines sont détaillées dans la Section 2.2). La transcription et les sous-titres sont globalement plus simples pour les émissions de stock (le score de lisibilité FRE est plus élevé pour la transcription et les sous-titres stock) : cela correspond notamment au fait que certaines de ces émissions contiennent moins d'interventions spontanées, qui forment des énoncés moins structurés et plus longs (les pauses étant plus fréquentes et plus appuyées dans les discours préparés). La différence de FRE (nettement plus grande pour le direct) et le taux de compression (sensiblement plus faible pour le direct) entre la transcription et les sous-titres suggèrent une simplification plus importante dans le cas du direct. Cependant, la distance d'édition normalisée et le score BLEU montrent que les sous-titres sont plus proches de la transcription pour le direct que pour le stock. En fait, bien qu'opérant davantage de suppressions de mots (par exemple sur des marques de l'expression orale telles que les hésitations ou les répétitions), les sous-titres produits en direct procèdent à relativement moins de réécriture que ceux en stock. Il apparaît également qu'en dépit de la compression plus forte, les sous-titres produits en direct sont de façon générale plus denses (le nombre caractères par ligne (CPL) et le nombre de caractères par seconde (CPS) sont légèrement plus faibles pour le stock, et la recommandation de 15 car/s est bien moins souvent respectée dans les sous-titres en direct).

3. Voir www.vocapia.com/speech-to-text-technology.html.

TR	Tout au long de la journée, des orages violents, de fortes pluies et quelles conséquences pour la population, faisons le point ce soir sur cette soudaine montée des eaux et sur les vents violents qui ont soufflé cet après-midi, dans les Bouches-du-Rhône à Marignane et je vous le disais sur la Côte-d'Azur à à Valbonne Vence ou encore à Nice, Alexandre Christophe Larocca.
ST	Des orages violents, de fortes pluies et quelles conséquences pour <p> la population ? <p> Faisons le point sur cette soudaine montée des eaux et sur les vents <p> violents qui ont soufflé cet après-midi... <p>

TABLE 1 – Exemple de segment transcrit automatiquement TR (source) et de segment sous-titre ST (cible) produit par un sous-titreur professionnel dans les conditions du direct. Les balises représentent la segmentation à l'affichage : `<br>` pour un saut de ligne au sein d'un bloc, `<p>` pour une fin de bloc (et changement d'écran).



(a) Métriques de simplification direct / stock

(b) Respect des normes de sous-titrage direct / stock

FIGURE 1 – Comparaison direct / stock au sein du corpus, selon plusieurs mesures portant sur la transcription (TR) ou les sous-titres (ST). NED et CR sont respectivement la distance d'édition normalisée et le taux de compression entre la transcription et les sous-titres (calculés au niveau caractère, et moyennés sur les segments). Les autres métriques sont décrites à la Section 2.2.

2.2 Métriques

Qualité et simplicité des phrases

BLEU (Papineni et al., 2002) est une métrique standard pour la traduction automatique. Xu et al. (2016) ont montré que dans le cas de la simplification, BLEU corrèle les jugements humains pour le sens et la grammaticalité, mais pas pour la simplicité. Nous utilisons l'implantation *SacreBLEU* de Post (2018).

SARI (Xu et al., 2016) compare les opérations d'édition (insertion, copie, suppression de n-gramme) observées entre l'entrée et la sortie, avec celles observées entre l'entrée et les références [4]. Nous utilisons l'implantation de la bibliothèque *EASSE* (Alva-Manchego et al., 2019).

Flesch Reading Ease (FRE) (Flesch, 1948) évalue la lisibilité, en se fondant sur le nombre moyen de mots par phrase et sur le nombre moyen de syllabes par mot. Nous reprenons la formule adaptée au français par Kandel & Moles (1958).

4. N'ayant qu'une seule version de sous-titres pour les émissions, nous ne mesurons SARI qu'avec une référence.

Respect des normes superficielles de sous-titrage

L'affichage de sous-titres nécessite des informations précisant certains aspects de la présentation à l'écran, tels que la segmentation du texte en blocs et en lignes, le temps d'apparition de chaque bloc, la couleur des caractères, ou encore le positionnement horizontal des lignes. Ce formatage doit se conformer à des codes et des normes qui assurent la lisibilité des sous-titres.

Le nombre de caractères par lignes (*CPL*) et le nombre de caractères par seconde (*CPS*, calculé à partir de la durée d'affichage des blocs) sont en particulier soumis à des recommandations. Pour rendre compte du respect de ces contraintes, nous calculons la proportion de lignes dont la longueur dépasse 36 car, *CPL>36*, ainsi que la proportion de blocs qui dépassent une fréquence d'affichage de 15 car/s, *CPS>15* (ces seuils correspondent à des valeurs de référence).

Qualité de la segmentation des sous-titres

Nous reprenons deux métriques proposées respectivement par Matusov et al. (2019) et Karakanta et al. (2020a) pour évaluer la segmentation des sous-titres :

— Nous calculons *BLEU* en conservant les balises de fin de ligne et de fin de bloc dans les prédictions et les références. Cette mesure, que nous notons *BLEU-br*, permet d'évaluer indirectement le positionnement des balises de sous-titrage dans les phrases.

— Nous calculons le score *TER* (Snover et al., 2006) entre la sortie du système et la référence en masquant tous les mots à l'exception des balises de segmentation `<p>` et `<br>`.

Précision du contrôle de longueur

Pour estimer la précision du contrôle de longueur (opéré par les méthodes **LRPE** et **LDPE**, voir Section 3.2), nous avons choisi de calculer l'*erreur absolue moyenne* (EAM) des taux de compression obtenus par rapport aux taux de compression visés :

$$EAM = \frac{1}{n}\sum_{i=1}^{n}|\hat{r}_i - r_i|, \tag{1}$$

où n est la taille de l'ensemble de test, et $\hat{r}_i$ et r_i sont respectivement le taux de compression obtenu et le taux de compression visé pour la i-ème phrase.

L'*erreur absolue* (EA) $|\hat{r} - r|$ peut aussi être vue comme la différence entre la longueur produite et la longueur visée $|l_{\hat{y}} - r \times l_x|$ rapportée à la longueur source l_x. Pour compléter nos métriques, nous avons évalué la proportion d'instances pour lesquelles l'erreur absolue est inférieure à 10 %.

3 Méthodes

Les systèmes de production de sous-titres que nous évaluons ont pour entrée des phrases issues de la transcription automatique des paroles prononcées dans une émission. Nous prenons comme référentiel un système qui conserve telle quelle la transcription mot-à-mot réalisée par l'outil de reconnaissance de parole (ce système est noté **Source** dans le Tableau 2). Nos modèles réalisent

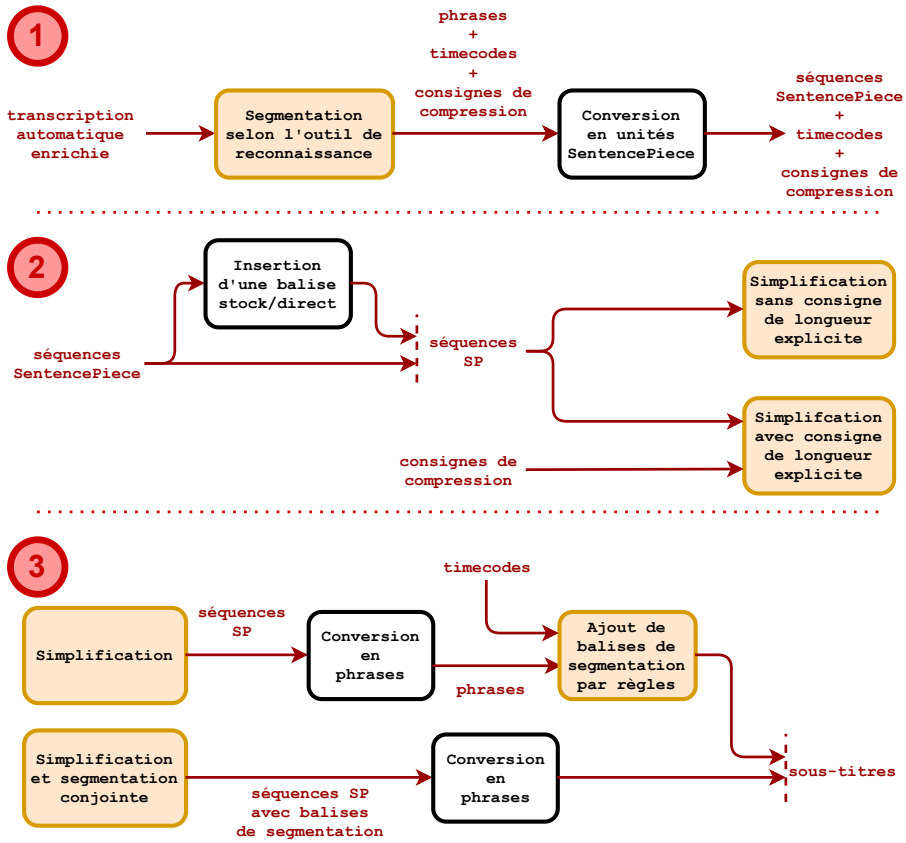


FIGURE 2 – Architecture pour le sous-titrage monolingue

une simplification via l'architecture *Transformer* (Vaswani et al., 2017). Nous expérimentons en plus l'emploi de mécanismes de contrôle de longueur, et l'intégration dans les données de balises pour la segmentation à l'affichage des sous-titres, ou la qualification du type d'émission. Nous pré-traitons les données en utilisant *Sentencepiece* (Kudo & Richardson, 2018), avec un vocabulaire de 16 000 unités.

L'architecture globale des modèles est représentée sur la figure 2.

3.1 Modèles *Transformer* pour la simplification

Pour réaliser la simplification de la transcription nous avons utilisé des modèles à base de *Transformer*, en ré-implémentant l'architecture de Vaswani et al. (2017) (dimension de plongement $d_{\text{model}} = 256$, dimension du perceptron multicouche $d_{\text{ff}} = 1024$, nombre de couches pour l'encodeur / le décodeur $N = 6$, nombre de têtes d'attention $h = 8$). L'optimisation a été faite avec *Adam* (Kingma & Ba, 2015) (avec $\beta_1 = 0,9$, $\beta_2 = 0,98$, $eps = 10^{-9}$). Nous avons également repris la méthode de variation du taux d'apprentissage proposée par Vaswani et al. (2017), en fixant le nombre d'étapes d'échauffement à 4000. Les modèles ont été entraînés sur 265 000 paires de phrases jusqu'à ce que la fonction de perte n'ait pas augmenté pendant 5 époques.

3.2 Contrôle de la longueur

Vaswani et al. (2017) ont défini un encodage positionnel, qui dans le *Transformer* est combiné avec le plongement de chaque mot de la phrase d'entrée (dans la partie encodeur) ou de l'amorce de phrase produite (dans la partie décodeur) :

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right), \tag{2}$$

où pos est la position du mot dans la phrase, et $2i$ (resp. $2i+1$) correspond aux dimensions paires (resp. impaires) de l'encodage. Les modèles avec encodage classique sont notés $\mathbf{Transf}$ (Tableau 2).

Pour contrôler la longueur de la sortie de certains de nos modèles, nous avons ré-implémenté les variantes $\mathbf{LRPE}$ et $\mathbf{LDPE}$ proposées par (Takase & Okazaki, 2019). Ces encodages intègrent une consigne sur la longueur à atteindre l, par ratio ($\mathbf{LRPE}$) ou différence ($\mathbf{LDPE}$) avec la position pos :

$$LRPE_{(pos,l,2i)} = \sin\left(\frac{pos}{l^{2i/d_{model}}}\right), \quad LRPE_{(pos,l,2i+1)} = \cos\left(\frac{pos}{l^{2i/d_{model}}}\right), \tag{3}$$

$$LDPE_{(pos,l,2i)} = \sin\left(\frac{l-pos}{10000^{2i/d_{model}}}\right), \quad LDPE_{(pos,l,2i+1)} = \cos\left(\frac{l-pos}{10000^{2i/d_{model}}}\right). \tag{4}$$

l est égal à la longueur de la séquence cible de référence pendant la période d'entraînement, mais est fixé par l'utilisateur pendant la période de test. $\mathbf{LRPE}$ caractérise à la fois la position courante pos et la longueur totale voulue l, tandis que $\mathbf{LDPE}$ exprime une distance à l'objectif de longueur.

Dans nos expériences, nous avons modulé les objectifs de longueur afin de contraindre les modèles $\mathbf{LRPE}$ et $\mathbf{LDPE}$ à générer des phrases respectant soit un taux de compression constant r (auquel cas l est égale à la longueur de la phrase d'entrée multipliée par r), soit une fréquence d'affichage des caractères constante f (auquel cas l est égale à la durée allouée à l'affichage des tronçons de la phrase multipliée par f).

3.3 Intégration de balises

Dans notre processus de production de sous-titres, le découpage temporel est effectué à partir des périodes de parole identifiées par l'outil de reconnaissance de parole (en permettant à l'affichage de durer quelques secondes supplémentaires pendant les éventuels silences).

Concernant le découpage spatial, notre solution de base est un système à règles implémentant une heuristique simple, qui produit des tronçons de phrases dont la longueur appartient à un intervalle jugé acceptable, et qui favorise la segmentation au niveau des ponctuations. Nous avons appliqué cette méthode avec le référentiel conservant la transcription, et avec la sortie d'un modèle *Transformer* sans contrôle de longueur. Les systèmes résultants sont respectivement notés $\mathbf{Source_R}$ et $\mathbf{Transf_R}$.

L'autre méthode mise en place consiste à intégrer des balises aux emplacements des coupures dans les sous-titres utilisés pour l'apprentissage, comme dans l'exemple du tableau 1. Les systèmes utilisant cette méthode réalisent conjointement la simplification et la segmentation, et sont notés $\mathbf{Transf_B}$, $\mathbf{LRPE_B}$ et $\mathbf{LDPE_B}$.

Enfin, en suivant la littérature sur la génération de phrases contrôlée (Sennrich et al., 2016; Kobus

et al., 2017; Martin et al., 2020), nous avons entraîné un modèle *Transformer* ($\mathbf{Transf_{BT}}$ [5]) en ajoutant au début des phrases sources une balise spécifique qui indique si la phrase cible attendue est un sous-titre de stock ou de direct.

4 Résultats

Le tableau 2 présente les résultats de l'évaluation que nous avons réalisée sur un ensemble de 10 vidéos d'émissions représentatives des programmes traités, pour une durée cumulée d'environ 10 h. Les segments sous-titres de références ont été constitués par alignement automatique avec les phrases de la transcription automatique [6].

Il apparaît que les mécanismes de contrôle de longueur n'améliorent pas la qualité de la simplification, les modèles *Transformer* à encodage positionnel classique obtenant des scores BLEU et SARI meilleurs. Les scores TER-br légèrement inférieurs semblent néanmoins indiquer que **LRPE** et **LDPE** permettent un meilleur positionnement des coupures dans les phrases. Ces modèles respectent aussi de façon plus régulière la norme sur la fréquence d'affichage des caractères (en particulier lorsque l'objectif de longueur est modulé pour suivre une fréquence constante).

L'ajout d'une balise pour spécifier le type d'émission (stock ou direct) semble être bénéfique pour la segmentation, dans la mesure où $\mathbf{Transf_{BT}}$ est meilleur que $\mathbf{Transf_{B}}$ pour TER-br, CPL>36 et CPS>15, tout en étant comparable par ailleurs.

La précision du contrôle de longueur est relative, puisque la différence entre la consigne de longueur et sa réalisation représente en moyenne entre 16 et 20 % de la longueur source (EAM). **LRPE** et **LDPE** sont ici comparables du point de vue de l'effectivité de ce contrôle. Concernant la qualité des phrases produites (SARI, BLEU, BLEU-br), **LRPE** est supérieur à **LDPE**, et la poursuite d'une fréquence de caractères constante semble préférable à l'application d'un unique taux de compression (ce qui paraît effectivement plus proche de ce que ferait un sous-titreur humain).

Afin d'estimer l'importance dans les résultats des erreurs liées à la reconnaissance automatique de parole, nous avons fait réaliser une transcription manuelle (professionnelle) des émissions de notre ensemble de test, et avons évalué certains de nos modèles à partir de cette transcription considérée comme une version « idéale » de la transcription automatique. Nous observons dans ces cas un gain substantiel pour le score BLEU (entre 1,5 et 2 points), mais une baisse de SARI (d'entre 2 et 2,5 points) : les défauts dans la transcription automatique pourraient pousser les systèmes à réaliser certaines simplifications pour produire des phrases plausibles.

À titre de comparaison, Gangi et al. (2019) obtiennent des scores BLEU de l'ordre de 30 sur une traduction anglais-italien à partir d'une transcription automatique, et Matusov et al. (2019) donnent des scores BLEU-br de l'ordre de 40 pour un documentaire et 30 pour une sitcom, pour un sous-titrage multilingue de l'anglais en espagnol (ces tâches sont néanmoins plus difficiles que la nôtre).

Une première observation qui se dégage du Tableau 3 est la grande variabilité des scores selon les émissions. Les variations sont similaires pour les différents modèles : les meilleurs résultats (pour la qualité des phrases au moins) sont obtenus sur la catégorie *journal*, et les moins bons sur la catégorie *jeu*, la catégorie *magazine* étant intermédiaire. Ces écarts s'expliquent en partie par la qualité de la

5. Les balises de segmentation étaient également utilisées avec ce système.

6. Une partie des phrases transcrites (représentant dans l'ensemble $\sim$ 6 % des mots) n'ont pas pu être alignées avec les phrases des sous-titres ; nous avons décidé de les écarter pour l'évaluation.

Modèle	BLEU-br	BLEU	SARI	TER-br	CPL>36	CPS>15	EAM	EA<10 %
Source_R	27,5	34,3	18,1	0,608	0 %	83,1 %	-	-
Cible	100	100	100	0	0 %	46,2 %	-	-
Transf_R	38,1	43,3	52,2	0,390	0 %	63,5 %	-	-
Transf_B	41,5	43,8	52,9	0,381	6,1 %	63,8 %	-	-
Transf^*_B	43,2	46,0	50,5	0,360	6,1 %	53,7 %	-	-
$\text{LRPE}_{B;r=0,75}$	35,6	35,9	49,8	0,351	5,5 %	16,8 %	15,7 %	18,9 %
$\text{LRPE}_{B;f=14,5}$	38,7	39,5	51,2	0,313	5,3 %	1,2 %	20,2 %	28,5 %
$\text{LRPE}^*_{B;f=14,5}$	39,9	41,1	49,3	0,317	5,3 %	1,3 %	21,1 %	25,7 %
$\text{LDPE}_{B;r=0,75}$	34,5	35,2	49,5	0,351	7,0 %	16,3 %	16,5 %	16,3 %
$\text{LDPE}_{B;f=14,5}$	37,3	38,7	50,6	0,316	7,6 %	0,8 %	20,4 %	27,4 %
Transf_{BT}	41,6	43,9	53,1	0,375	4,1 %	62,2 %	-	-
Transf^*_{BT}	42,6	45,3	50,6	0,364	4,1 %	57,6 %	-	-

TABLE 2 – Résultats de l'évaluation des modèles sur un groupe d'émissions de test. Les métriques EAM et EA<10 % ne sont testées que pour les systèmes qui intègrent des objectifs de longueur. Les évaluations de modèles utilisant une version de référence des transcriptions réalisée manuellement sont notées par (*).

transcription automatique : les taux d'erreur par mot (*Word Error Rate*) par rapport à la transcription manuelle ont été estimés à entre 10 et 40 % en fonction des émissions (les journaux obtenant les taux les plus bas, et les jeux les taux les plus hauts). La reconnaissance vocale est notamment affectée par le débit de parole, la clarté de la prononciation, les dialogues avec recouvrement, et généralement la présence de bruits parasites. Par exemple, le jeu télévisé choisi dans notre test contient beaucoup de séquences avec de la musique ou des rires, et des échanges rapides. De plus, la nature du programme fait qu'une partie des phrases ont une structure assez spécifique (énoncé d'une question de culture générale, ou réponse très courte d'un candidat). Nous remarquons enfin que le respect de la norme CPS (fortement lié au débit de parole initial) change significativement d'une émission à l'autre, notamment dans les sous-titres de référence.

Le tableau 4 présente des exemples de transformations réalisées par les systèmes $\mathbf{Transf_B}$ et $\mathbf{LRPE_{B;f=14,5}}$. Nous observons que les modèles ont appris à re-segmenter les phrases (après « émission » dans le premier exemple, « mâts » dans le deuxième, et « majoritaires » dans le troisième), et à reconnaître la forme interrogative (premier exemple). Occasionnellement, ils peuvent également corriger des erreurs de grammaire (« avait », premier exemple). Les conventions orthographiques et typographiques, telles que l'écriture de « % » (troisième exemple) sont globalement gérées efficacement. Le deuxième exemple montre que le modèle $\mathbf{Transf_B}$ élague trop la phrase initiale par rapport à la référence. $\mathbf{LRPE_{B;f=14,5}}$ fait mieux dans ce cas, bénéficiant de l'information de longueur en rapport avec le temps d'affichage disponible. Toutefois, la contrainte de longueur induit souvent des suppressions abruptes, provoquant la perte du sens original (troisième exemple) ou de la cohérence syntaxique. Outre l'abandon d'éléments importants, les erreurs fréquemment commises par les modèles comptent de mauvaises segmentations, et la conservation d'artefacts de la transcription automatique.

En alignant les phrases produites par $\mathbf{Transf_B}$ avec les références de notre ensemble de test, nous avons noté que les opérations d'édition les plus fréquentes étaient les suppressions de mots connecteurs ou introductifs : « et », « que », « c'est », « ça », « donc », « qui »...

Modèle	BLEU-br	BLEU	SARI	FRE	TER-br	CPL>36	CPS>15
Jeu (stock)							
Source_R	24,1	24,0	14,6	68,6	0,68	0 %	74,4 %
Cible	100	100	100	94,2	0	0 %	28,9 %
Transf_B	37,1	31,9	48,7	94,5	0,41	4,7 %	46,9 %
$\text{LRPE}_{B;f=14,5}$	37,5	31,1	48,0	95,1	0,33	4,0 %	4,3 %
Journal (direct)							
Source_R	36,4	49,0	23,0	59,3	0,36	0 %	81,1 %
Cible	100	100	100	82,1	0	0 %	62,3 %
Transf_B	45,6	56,9	57,5	83,2	0,28	5,9 %	72,6 %
$\text{LRPE}_{B;f=14,5}$	40,1	48,4	54,0	84,1	0,29	7,9 %	0,8 %
Magazine (stock)							
Source_R	29,0	33,4	18,3	68,9	0,42	0 %	64,7 %
Cible	100	100	100	94,3	0	0 %	21,0 %
Transf_B	45,8	45,0	52,9	93,3	0,29	5,7 %	43,7 %
$\text{LRPE}_{B;f=14,5}$	41,5	40,4	51,3	94,1	0,25	5,5 %	0,4 %

TABLE 3 – Résultats par émission de l'évaluation des modèles.

5 Travaux connexes

Les besoins d'accessibilité audio-visuelle ainsi que l'exportation de programmes vidéos ont depuis un certain temps suscité un intérêt pour l'automatisation du sous-titrage (Daelemans et al., 2004; Koponen et al., 2020). La procédure fondée sur la mise en cascade d'un système de reconnaissance de parole et d'un système de simplification / compression a longtemps été privilégiée pour le sous-titrage intralingue (Gangi et al., 2019). Récemment, les progrès dans la direction de la traduction de parole sans transcription intermédiaire (Bérard et al., 2016) ont permis l'émergence d'approches bout-en-bout pouvant prendre en compte les indices prosodiques (Karakanta et al., 2020a).

La simplification et la compression de phrases ont abondamment été étudiées pour leur multiples applications, parmi lesquelles la production de sous-titres. Ces tâches ont particulièrement été abordées en reprenant les méthodes de la traduction automatique, des systèmes à règles (Cohn & Lapata, 2008) aux modèles neuronaux (Zhang & Lapata, 2017; Dong et al., 2019).

Pour le sous-titrage automatique, la question du contrôle de la longueur des phrases engendrées est essentielle du fait des contraintes spatiales et temporelles (Angerbauer et al., 2019). Kikuchi et al. (2016) ont mis en place un tel contrôle en introduisant des consignes de longueur dans les états cachés d'un RNN. Takase & Okazaki (2019) ont adapté cette idée à l'architecture *Transformer*, en tirant parti de l'encodage positionnel pour intégrer l'objectif de longueur.

L'ajout d'une étiquette spécifique en début de séquence afin de qualifier un attribut attendu dans la phrase de sortie rejoint une riche littérature de production contrôlée de texte, cette stratégie ayant notamment mise en œuvre pour la formalité (Sennrich et al., 2016), le domaine (Kobus et al., 2017) ou encore la longueur (Lakew et al., 2019; Martin et al., 2020).

TR	**Suite à** votre passage dans **l'émission** comment les élèves à l'époque **avait réagi.**
Transf_B	Suite à votre passage dans **l'émission,** comment les élèves <p> **avaient réagi ?** <p>
$\text{LRPE}_{B;f}$	Suite à votre passage dans **l'émission.** <p> Comment ? <p>
ST	**A** votre passage dans **l'émission,** comment les élèves **avaient réagi ?** <p>
TR	**Donc nous** on monte les mâts **oui mais bon, on va laisser finir**.
Transf_B	On monte les mâts. <p>
$\text{LRPE}_{B;f}$	On monte les mâts. <p> On va laisser finir. <p>
ST	**Nous,** on monte les mâts. <p> **Ah oui.** <p> **Stop.** On va **les** laisser finir. <p>
TR	En Alabama, les anti-avortement sont **majoritaires, 70 pourcent** des habitants se déclarent pour **une interdiction de l'IVG**.
Transf_B	En Alabama, les anti-avortement sont **majoritaires.** <p> **70 %** des habitants se déclarent pour **une interdiction de l'IVG**. <p>
$\text{LRPE}_{B;f}$	En Alabama, les anti-avortement sont **majoritaires.** <p> **70 %** des habitants se déclarent pour **une IVG**. <p>
ST	En Alabama, les anti-avortement sont **majoritaires.** <p> **70%** des habitants se déclarent pour **une interdiction de l'IVG**. <p>

TABLE 4 – Exemples de phrase engendrées par les modèles $\mathbf{Transf_B}$ et $\mathbf{LRPE_{B;f=14,5}}$, comparées à la transcription initiale (TR) et au sous-titre de référence (ST).

6 Conclusion

Nous avons présenté un nouveau corpus pour le sous-titrage automatique, et en avons fait une première utilisation en comparant différentes stratégies pour la production de sous-titres segmentés en vue de l'affichage sur un écran. Dans nos premiers essais, l'implémentation d'un contrôle de longueur améliore le respect de certaines normes superficielles, mais diminue la qualité de la simplification. L'ajout de balises caractérisant le type d'émission pour lequel sont engendrés les sous-titres semble être modérément bénéfique. Compte tenu des variations de résultats observées entre émissions, il pourrait être intéressant à l'avenir d'utiliser des balises pour des catégories d'émissions plus spécifiques. L'apprentissage pourrait être renforcé en intégrant davantage de données, provenant de ressources accessibles publiquement (MOOC ou TedTalk par exemple), ou créées artificiellement à partir de sous-titres sans transcription parallèle (une pseudo-transcription peut être engendrée automatiquement, par rétro-traduction notamment). Enfin il est envisageable de tester des architectures alternatives, adaptées à la traduction monolingue, telles que *Levenshtein Transformer* (Gu et al., 2019), ou encore *Pointer Networks* (Vinyals et al., 2015).

Remerciements

Nous remercions J.-L. Gauvain (LISN, CNRS) pour son aide dans la mise en œuvre des systèmes de transcription automatique, et E. Florence (france.tv access) pour l'accès aux données des émissions. Ce travail a bénéficié de calculs réalisés sur la plateforme LabIA. Ces travaux ont été menés dans le cadre du projet "Rosetta - RObot de Sous-titrage Et Toute Traduction Adaptés", financé par le Programme d'Investissements d'Avenir "Grands défis du numérique" de la Banque Publique d'Investissement (BPI).

Références

ALVA-MANCHEGO F., MARTIN L., SCARTON C. & SPECIA L. (2019). EASSE : easier automatic sentence simplification evaluation. CoRR, **abs/1908.04567**.

ANGERBAUER K., ADEL H. & VU N. T. (2019). Automatic compression of subtitles with neural networks and its effect on user experience. In G. KUBIN & Z. KACIC, Éds., Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019, p. 594–598 : ISCA. DOI : 10.21437/Interspeech.2019-1750.

BAHDANAU D., CHO K. & BENGIO Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. BENGIO & Y. LECUN, Éds., 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

BÉRARD A., PIETQUIN O., BESACIER L. & SERVAN C. (2016). Listen and Translate : A Proof of Concept for End-to-End Speech-to-Text Translation. In NIPS Workshop on end-to-end learning for speech and audio processing, Barcelona, Spain. HAL : hal-01408086.

COHN T. & LAPATA M. (2008). Sentence compression beyond word deletion. In Proceedings of the 22nd International Conference on Computational Linguistics, (COLING 2008), p. 137–144, Manchester, UK : Coling 2008 Organizing Committee.

DAELEMANS W., HÖTHKER A. & TJONG KIM SANG E. (2004). Automatic sentence simplification for subtitling in Dutch and English. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal : European Language Resources Association (ELRA).

DONG Y., LI Z., REZAGHOLIZADEH M. & CHEUNG J. C. K. (2019). EditNTS : An neural programmer-interpreter model for sentence simplification through explicit editing. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, p. 3393–3402, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1331.

FLESCH R. (1948). A new readability yardstick. Journal of applied psychology, **32**(3), 221.

GANGI M. A. D., ENYEDI R., BRUSADIN A. & FEDERICO M. (2019). Robust neural machine translation for clean and noisy speech transcripts. CoRR, **abs/1910.10238**.

GU J., WANG C. & ZHAO J. (2019). Levenshtein transformer. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ-BUC, E. FOX & R. GARNETT, Éds., Advances in Neural Information Processing Systems, volume 32, p. 11181–11191 : Curran Associates, Inc.

KANDEL L. & MOLES A. (1958). Application de l'indice de Flesch à la langue française. Cahiers Etudes de Radio-Télévision, **19**(1958), 253–274.

KARAKANTA A., NEGRI M. & TURCHI M. (2020a). Is 42 the answer to everything in subtitling-oriented speech translation ? In Proceedings of the 17th International Conference on Spoken Language Translation, p. 209–219, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.iwslt-1.26.

KARAKANTA A., NEGRI M. & TURCHI M. (2020b). MuST-cinema : a speech-to-subtitles corpus. In Proceedings of the 12th Language Resources and Evaluation Conference, p. 3727–3734, Marseille, France : European Language Resources Association.

KIKUCHI Y., NEUBIG G., SASANO R., TAKAMURA H. & OKUMURA M. (2016). Controlling output length in neural encoder-decoders. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, p. 1328–1338 : Association for Computational Linguistics. DOI : 10.18653/v1/D16-1140.

KINGMA D. P. & BA J. (2015). Adam : A method for stochastic optimization. In Y. BENGIO & Y. LECUN, Éds., 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

KOBUS C., CREGO J. & SENELLART J. (2017). Domain control for neural machine translation. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, p. 372–378, Varna, Bulgaria. DOI : 10.26615/978-954-452-049-6_049.

KOPONEN M., SULUBACAK U., VITIKAINEN K. & TIEDEMANN J. (2020). MT for subtitling : Investigating professional translators' user experience and feedback. In Proceedings of 1st Workshop on Post-Editing in Modern-Day Translation, p. 79–92, Virtual : Association for Machine Translation in the Americas.

KUDO T. & RICHARDSON J. (2018). SentencePiece : A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations, p. 66–71, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-2012.

LAKEW S. M., GANGI M. D. & FEDERICO M. (2019). Controlling the output length of neural machine translation. In Proceedings of IWSLT'2019.

MARTIN L., DE LA CLERGERIE É., SAGOT B. & BORDES A. (2020). Controllable sentence simplification. In Proceedings of the 12th Language Resources and Evaluation Conference, p. 4689–4698, Marseille, France : European Language Resources Association.

MATUSOV E., WILKEN P. & GEORGAKOPOULOU Y. (2019). Customizing neural machine translation for subtitling. In Proceedings of the Fourth Conference on Machine Translation (Volume 1 : Research Papers), p. 82–93, Florence, Italy. DOI : 10.18653/v1/W19-5209.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). BLEU : a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, p. 311–318, Stroudsburg, PA, USA : Association for Computational Linguistics. DOI : http ://dx.doi.org/10.3115/1073083.1073135.

POST M. (2018). A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation : Research Papers, p. 186–191, Belgium, Brussels : Association for Computational Linguistics.

RUSH A. M., CHOPRA S. & WESTON J. (2015). A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, p. 379–389 : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1044.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Controlling politeness in neural machine translation via side constraints. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, p. 35–40, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1005.

SNOVER M., DORR B., SCHWARTZ R., MICCIULLA L. & MAKHOUL J. (2006). A study of translation edit rate with targeted human annotation. In Proceedings of association for machine translation in the Americas, volume 200 : Cambridge, MA.

TAKASE S. & OKAZAKI N. (2019). Positional encoding to control output sequence length. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers), p. 3999–4004, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1401.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. VON LUXBURG, S. BENGIO, H. M. WALLACH, R. FERGUS, S. V. N. VISHWANATHAN & R. GARNETT, Éds., Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, p. 6000–6010.

VINYALS O., FORTUNATO M. & JAITLY N. (2015). Pointer networks. In C. CORTES, N. LAWRENCE, D. LEE, M. SUGIYAMA & R. GARNETT, Éds., Advances in Neural Information Processing Systems, volume 28, p. 2692–2700 : Curran Associates, Inc.

XU W., NAPOLES C., PAVLICK E., CHEN Q. & CALLISON-BURCH C. (2016). Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics, **4**, 401–415.

ZHANG X. & LAPATA M. (2017). Sentence simplification with deep reinforcement learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, p. 584–594, Copenhagen, Denmark : Association for Computational Linguistics.

ZHANG Y., YE Z., FENG Y., ZHAO D. & YAN R. (2017). A constrained sequence-to-sequence neural model for sentence simplification. CoRR, **abs/1704.02312**.

Deuxième partie

Articles courts

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Analyse en dépendances du français avec des plongements contextualisés

Loïc Grobol[1, 2, 3] Benoît Crabbé[1]
(1) LLF, CNRS, Université de Paris, 8, Rue Albert Einstein 75013 Paris, France
(2) Lattice, CNRS, ENS, PSL, Université Sorbonne Nouvelle, 1 Rue Maurice Arnoux, 92120 Montrouge, France
(3) LIFO, ICVL, Université d'Orléans, 45000 Orléans, France
loic.grobol@ens.psl.eu, benoit.crabbe@linguist.univ-paris-diderot.fr

RÉSUMÉ
Cet article présente un analyseur syntaxique en dépendances pour le français qui se compare favorablement à l'état de l'art sur la plupart des corpus de référence. L'analyseur s'appuie sur de riches représentations lexicales issues notamment de BERT et de FastText. On remarque que les représentations lexicales produites par FlauBERT ont un caractère auto-suffisant pour réaliser la tâche d'analyse syntaxique de manière optimale.

ABSTRACT
French dependency parsing with contextualized embeddings

This paper presents a dependency parser for French that compares favorably to the state of the art on several corpora. The parser relies on rich lexical representations from BERT and FastText. We notice that the lexical representations produced by FlauBERT are somehow self-sufficient to perform the syntax analysis task in an optimal way.

MOTS-CLÉS : Analyse syntaxique en dépendances du français, BERT, FastText.

KEYWORDS : Dependency Parsing, Parsing French, BERT, FastText.

1 Introduction

Cet article décrit un modèle d'analyse syntaxique en dépendances couplé à un étiqueteur morphosyntaxique pour le français [1]. Nous étudions plus spécifiquement l'impact de représentations lexicales apprises de manière non supervisée sur de gros volumes de texte telles que FastText et BERT en complément de plongements lexicaux standards. Nous commençons par expliciter le modèle d'analyse en section 2 avant de présenter les expériences sur les représentations lexicales en section 3.

2 Modèle d'analyse

Le modèle d'analyse combine à la fois un étiqueteur morphosyntaxique, un analyseur basé sur l'algorithme de Dozat & Manning (2017) et l'utilisation de riches représentations lexicales (Bojanowski *et al.*, 2017 ; Devlin *et al.*, 2019).

1. Disponible à https://github.com/bencrabbe/npdependency

Dans ce qui suit, on suppose qu'une phrase $w_1 ... w_n$ est représentée par une séquence de plongements lexicaux $\boldsymbol{X} = \boldsymbol{x}_1 ... \boldsymbol{x}_n$. En préalable à l'analyse nous traitons les plongements lexicaux à l'aide d'un LSTM pour construire une séquence de représentations contextualiséss $\boldsymbol{C} = \boldsymbol{c}_1 ... \boldsymbol{c}_n$:

$$\boldsymbol{C} = \text{LSTM}(\boldsymbol{X})$$

L'étiquetage morphosyntaxique est réalisé à l'aide d'un réseau à propagation avant (MLP) et d'une sortie SOFTMAX :

$$\hat{\boldsymbol{e}}_i = \textsc{softmax}(\text{MLP}^{\text{POS}}(\boldsymbol{c}_i))$$

$\hat{\boldsymbol{e}}_i$ est alors la distribution de scores des étiquettes morphosyntaxiques pour le mot w_i.

Pour l'analyse syntaxique, nous commençons par spécialiser les représentations de $\boldsymbol{C}$ de quatre manières différentes :

$$\begin{aligned}
\boldsymbol{h}_i^{\text{arc}-\text{dep}} &= \textsc{mlp}^{\text{arc}-\text{dep}}(\boldsymbol{c}_i) \\
\boldsymbol{h}_i^{\text{arc}-\text{gov}} &= \textsc{mlp}^{\text{arc}-\text{gov}}(\boldsymbol{c}_i) \\
\boldsymbol{h}_i^{\text{label}-\text{dep}} &= \textsc{mlp}^{\text{label}-\text{dep}}(\boldsymbol{c}_i) \\
\boldsymbol{h}_i^{\text{label}-\text{gov}} &= \textsc{mlp}^{\text{label}-\text{gov}}(\boldsymbol{c}_i)
\end{aligned}$$

où $\boldsymbol{h}_i^{\text{arc}-\text{gov}}$ (resp. $\boldsymbol{h}_i^{\text{arc}-\text{dep}}$) représente une spécialisation de la représentations de w_i du mot comme gouverneur (resp. comme dépendant). Ces représentations étant utilisées pour prédire les arcs, l'utilisation de représentations différentes pour les rôles de gouverneur et de dépendant permet d'éviter des effets de symétrie : le score d'un arc entre deux mots ne devrait en effet pas être le même selon que l'on choisisse soir l'un soit l'autre comme gouverneur. µ On réalise également une spécialisation analogue pour créer des représentations spécifiques pour la tâche d'étiquetage des arcs que l'on note $\boldsymbol{h}_i^{\text{label}-\text{dep}}$ et $\boldsymbol{h}_i^{\text{label}-\text{gov}}$.

La prédiction des arcs et de leurs étiquettes repose sur plusieurs fonctions biaffines de la forme :

$$\textsc{Biaff}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{W} \boldsymbol{y} + \boldsymbol{U}(\boldsymbol{x} \oplus \boldsymbol{y}) + \boldsymbol{b}$$

On utilise une première fonction biaffine pour attribuer un score $s_{j \to i}^{arc}$ à chaque arc entre un gouverneur w_j et un dépendant w_i :

$$s_{j \to i}^{arc} = \textsc{Biaff}(\boldsymbol{h}_i^{\text{arc}-\text{gov}}, \boldsymbol{h}_j^{\text{arc}-\text{dep}})$$

et autant de fonctions biaffines qu'il y a de types de dépendances ℓ pour attribuer un score à chaque étiquette possible pour ce même arc :

$$s_{j \to i}^{\ell} = \textsc{Biaff}_{\ell}(\boldsymbol{h}_i^{\text{label}-\text{gov}}, \boldsymbol{h}_j^{\text{label}-\text{dep}})$$

Ces fonctions diffèrent les unes des autres, les matrices de paramètres $\boldsymbol{W}_{\ell}$ et $\boldsymbol{U}_{\ell}$ étant *a priori* distinctes pour chaque ℓ.

Fonction objectif La fonction objectif du modèle est composée d'objectifs multiples. Le premier porte sur l'étiquetage morphosyntaxique. On suppose qu'une phrase est un couple (W, E) composé d'une séquence de mots $W = w_1 ... w_n$ et d'une séquence d'étiquettes de référence $E = e_1 ... e_n$. Cette première fonction calcule l'entropie croisée entre la distribution prédite par le modèle $\hat{\boldsymbol{e}}_i$ pour chaque tag et la distribution ponctuelle $\boldsymbol{e}_i = (\delta_{ij})_j$ qui encode e_i :

$$\mathcal{L}(W, T) = \sum_{i=1}^{n} H(\boldsymbol{e}_i, \hat{\boldsymbol{e}}_i)$$

Le second objectif porte sur la prédiction des arcs, indépendamment de leurs étiquettes. Dans ce cas on suppose que la phrase est un couple (W,D), où D est un ensemble d'arcs de dépendance de référence. Pour chaque arc $(j \rightarrow i) \in D$ on construit un vecteur $\boldsymbol{g}_i = (\delta jk)_k$ qui indique la position du gouverneur w_j du dépendant w_i. On obtient du modèle une matrice $\hat{\boldsymbol{G}} = \hat{\boldsymbol{g}}_{i,j}$ telle que pour tout i $(\hat{\boldsymbol{g}}_{i,j})_j = \textsc{softmax}(s^{arc}_{1\rightarrow i} ... s^{arc}_{n\rightarrow i})$ et la fonction objectif est alors de la forme suivante :

$$\mathcal{L}(W,D) = \sum_{i=1}^{n} H(\boldsymbol{g}_i, \hat{\boldsymbol{g}}_i)$$

Le dernier objectif concerne l'étiquetage des dépendances. Celui-ci suppose un étiquetage de l'ensemble des arcs de référence par une fonction L. On code l'étiquetage de référence d'un arc $(j \rightarrow i) \in D$ par une étiquette $\ell = L(j \rightarrow i)$ par le vecteur $\boldsymbol{l}_{j\rightarrow i} = (\delta_{k\ell})_k$. Pour chaque arc $(j \rightarrow i) \in D$ on obtient du modèle un vecteur de score $\hat{\boldsymbol{l}}_{j\rightarrow i} = \textsc{softmax}(s^{\ell_1}_{j\rightarrow i} ... s^{\ell_k}_{j\rightarrow i})$. Pour une phrase donnée on calcule l'entropie croisée, c'est-à-dire :

$$\mathcal{L}(W,D,L) = \sum_{(j\rightarrow i)\in D} H(\boldsymbol{l}_{j\rightarrow i}, \hat{\boldsymbol{l}}_{j\rightarrow i})$$

Considérons un corpus arboré $T = ((W_i, E_i, D_i, L_i))_{1 \leq i \leq N}$ annoté en dépendances et étiqueté morphosyntaxiquement. La fonction objectif globale est la somme des trois objectifs définis ci-dessus :

$$\mathcal{L}(T) = \sum_{i=1}^{N} \mathcal{L}(W_i, E_i) + \mathcal{L}(W_i, D_i) + \mathcal{L}(W_i, D_i, L_i)$$

Le problème d'optimisation consiste alors à minimiser $\mathcal{L}(T)$. En pratique, nous utilisons la variante de descente de gradient stochastique appelée Adam (Kingma & Ba, 2014) et retenons le modèle qui minimise la perte sur le jeu de validation au cours d'un nombre e d'epochs fixé à l'avance.

Prédiction Pour prédire un arbre de dépendance à partir d'une phrase $w_1 ... w_n$ donnée en entrée, on commence par évaluer $s^{arc}_{j\rightarrow i}$ pour $1 \leq i \leq n$ et $1 \leq j \leq n$. La matrice de scores $\hat{\boldsymbol{G}}$ est vue comme la matrice de poids d'un graphe pondéré complet pour lequel on calcule une arborescence A couvrante de poids maximal à l'aide de l'algorithme de Chu-Liu/Edmonds (Chu & Liu, 1965 ; Edmonds, 1967). On assigne alors à chaque arc $(j \rightarrow i) \in A$ l'étiquette de score maximal :

2.1 Représentations lexicales

L'analyseur utilise des représentations lexicales de natures diverses, et la représentation $\boldsymbol{x}_i$ d'un mot w_i dans la phrase est la concaténation de représentations calculés par différents modèles lexicaux :

$$\boldsymbol{x}_i = \textsc{FastText}(w_i) \oplus \textsc{BERT}(w_i) \oplus \textsc{char-rnn}(w_i) \oplus \textsc{lookup}(w_i) \quad (1)$$

La représentation LOOKUP est une représentation vectorielle ne dépendant que de la forme lexicale de w_i et stockée dans un dictionnaire. La représentation au niveau des caractères (CHAR-RNN) est obtenu par encodage de la séquence des caractères du mot à l'aide d'un bi-LSTM (Hochreiter & Schmidhuber, 1997). Les paramètres de ces deux représentations sont initialisées aléatoirement et appris sur les données du treebank d'entraînement en même temps que les paramètres de l'analyseur.

En revanche, les représentations FASTTEXT (Bojanowski *et al.*, 2017) et BERT (Devlin *et al.*, 2019) sont obtenus à partir de modèles entraînés sur de beaucoup plus gros volumes de données, et

sont seulement ajustés sur le treebank d'entrainement. L'embedding FastText est la moyenne des embeddings $\boldsymbol{x}_s$ de l'ensemble $S(w_i)$ des sous mots qui composent le mot w_i. Les embeddings BERT sont calculés à l'aide d'une succession de réseaux Transformer (Vaswani *et al.*, 2017). Précisément, on utilise une séquence de transformations de la forme :

$$\begin{aligned} \boldsymbol{c}_i^0 &= \textsc{lookup}(w'_i) \oplus \textsc{position}_i \\ \boldsymbol{c}_i^{l+1} &= \textsc{Transformer}(\boldsymbol{c}_i^l; \boldsymbol{c}^l) \qquad (0 \leq l < 12) \\ \text{BERT}(w'_i) &= \frac{1}{12} \sum_l \boldsymbol{c}_i^l \end{aligned}$$

où position est une famille de vecteurs encodant des positions et où $\boldsymbol{w}'$ n'est plus une séquence de mots, mais une séquence de *sous*-mots fournie par un segmenteur appris automatiquement Enfin, comme représentation $\text{BERT}(w_i)$ du mot w_i, on choisit la moyenne des embeddings de ses sous-mots.

3 Expériences

Nous avons testé l'analyseur sur différents jeux de données, principalement issues du projet Universal Dependencies (Zeman *et al.*, 2020). Les expériences de développement sont réalisées sur le sous-corpus de développement du corpus UD_French-GSD (Guillaume *et al.*, 2019), deuxième plus grand corpus de français disponible dans UD. Celles-ci permettent à mettre en perspective, par ablation, l'impact des différents modèles de représentation lexicales. Elles nous ont également permis d'optimiser les hyperparamètres de nos modèles dans ces différentes configurations et de la procédure d'apprentissage. Ces optimisations ont été faites empiriquement, le nombre et le coût de ces expériences rendant une recherche systématique peu envisageable. Les résultats données pour ces expériences le sont sur le jeu de données de développement et non sur celui de test, afin d'éviter des effets de surapprentissage d'architecture.

Finalement nous testons sur les autres principaux corpus UD pour le français (UD_French-Sequoia (Candito & Seddah, 2012 ; Bonfante *et al.*, 2018) et UD_French-Spoken (Lacheret *et al.*, 2014 ; Gerdes & Kahane, 2017)) à l'exception du French Treebank, pour lequel nous reprenons la version utilisée pour la campagne d'évaluation SPMRL 2013 (Seddah *et al.*, 2013), ceci afin de pouvoir nous comparer plus facilement aux travaux existants. Tous les corpus UD sont utilisés dans leur version 2.7.

Nous avons utilisé principalement le modèle FlauBERT (Le *et al.*, 2020) comme implémentation du modèle BERT. Pour la tâche d'analyse syntaxique nous avons remarqué lors d'expériences préliminaires qu'il se comporte généralement un peu mieux que le modèle CamemBERT (Martin *et al.*, 2020) même si les différences sont dans la marge d'erreur. Les modèles BERT utilisés sont les versions `base-cased` de FlauBERT et de mBERT tels que distribués dans la version 4.2.2 de la bibliothèque Transformers (Wolf *et al.*, 2020).

3.1 Expériences de développement

Le modèle d'analyse syntaxique décrit ici se caractérise essentiellement par un enrichissement des représentations lexicales acquises à partir de gros volumes de données. L'ajout de ces représentations a également un coût non négligeable sur la taille du modèle et sur les temps d'exécution. Nous

TABLE 1 – Résultats (dev) des expériences d'ablations sur UD_FRENCH-GSD 2.7

FASTTEXT	FLAUBERT	Lookup	Caractères	UPOS	UAS	LAS	CLAS
+	+	+	+	98,61	96,74	95,51	92,84
-	+	-	-	98,56	96,73	95,56	92,87
+	-	+	+	97,72	93,67	91,65	87,02
+	-	+	-	97,81	93,70	91,64	86,96
+	-	-	+	97,36	92,66	90,34	85,18
+	-	-	-	97,31	92,94	90,58	85,61
-	-	+	+	96,72	92,61	90,27	84,81
-	-	+	-	96,74	92,75	90,26	84,86
-	-	-	+	95,54	91,26	88,32	82,23

Lookup et caractères sont neutralisés en leur affectant des plongements de taille 2 (ce qui les rend moralement inexploitables par le modèle sans nécessiter un changement d'architecture), les plongements FLAUBERT sont neutralisés en les supprimant complètement des entrées. Enfin, pour les plongements FASTTEXT, − signifie que les plongements utilisés n'ont pas été préentraînés.
Les scores rapportés ici sont obtenus en répétant la procédure d'entraînement avec cinq germes aléatoires et en conservant les résultats du modèle le plus performant (en terme de LAS) sur le corpus de développement. Ces scores sont calculés par le script d'évaluation officiel de la campagne d'évaluation CoNLL 2018 [2] dont nous conservons [3] les métriques UPOS, UAS, LAS, et CLAS (Nivre & Fang, 2017).

reportons ici quelques chiffres sur les performances de notre architecture en suivant une méthode d'ablation sur le corpus de développement. Les deux premières lignes du tableau 1 apportent un élément de réponse à la question : peut on se contenter de BERT comme seule représentation lexicale pour le parsing ? Il semble que la réponse est affirmative vu que la différence avec un modèle où toutes les autres représentations sont neutralisées est négligeable.

Notre seconde question est : peut-on se passer de BERT et obtenir des résultats équivalents en analyse syntaxique ? La troisième ligne du tableau donne notre meilleur modèle qui n'utilise pas de représentations de type BERT et contribue à apporter une réponse négative à la question.

En revanche, ce modèle utilise FASTTEXT et on remarque à partir des lignes 3 à 6 de la table que l'ablation additionnelle des représentations LOOKUP semble avoir un impact nettement plus significatif que l'omission des embeddings de caractères. Remarquons finalement à partir des lignes 7, 8 et 9 qu'utiliser des représentations FASTTEXT non-préapprises entraîne une dégradation significative des performances. La suppression du sous modèle de caractères a là encore un effet négligeable.

Dans l'ensemble on remarque que l'apport de BERT est décisif : près de 5 points en LAS par rapport à un modèle appris uniquement à l'aide de représentations de mots sur le corpus d'entraînement. L'apport de FastText est plus modeste, et le sous-modèle de caractère semble avoir une contribution nulle dans les différentes configurations testées.

Ces tendances sont par ailleurs cohérentes pour les métriques UAS, LAS, mais aussi et surtout pour CLAS, ce qui suggère qu'elles sont bien liées à de réelles différences de performances pour la reconnaissance des structures syntaxiques à l'échelle de la phrase plutôt qu'à des différence dans

2. https://github.com/ufal/conll2018/tree/e865d82e4c296d1660c9e7efcceb79aa418b6348
3. Les métriques MLAS et BLEX, bien que plus récentes et préférées à CLAS dans cette campagne d'évaluation, n'auraient que peu de sens ici, puisque notre système ne prédit pas les traits morphologiques qu'elles mesurent.

TABLE 2 – Comparaisons entre les performances de nos modèles (sur différents corpus de test) en utilisant FLAUBERT, MBERT ou pas de représentations contextuelles (no BERT) et les performances de modèles à l'état de l'art.

(a) Résultats pour UD_FRENCH-GSD

Modèle	UPOS	UAS	LAS	CLAS
MBERT	*98,13*	93,95	92,08	88,00
FLAUBERT	**98,56**	95,67	**94,19**	**91,16**
no BERT	97,24	91,74	89,04	84,02
Martin *et al.*	*98,18*	—	92,57	—
Stanza	97,30	91,38	89,05	84,38
UD-Pipe 2	97,98	92,55	90,31	—

(b) Résultats pour FTB-SPRML

Modèle	UPOS	UAS	LAS
MBERT	*98,59*	91,68	88,48
FLAUBERT	**98,78**	**92,56**	**89,64**
no BERT	98,03	88,54	84,54
Le *et al.*	—	91,61	88,47
Constant *et al.*	—	89,19	85,86

(c) Résultats pour UD_FRENCH-SEQUOIA

Modèle	UPOS	UAS	LAS	CLAS
MBERT	*99,01*	94,26	92,75	90,02
FLAUBERT	**99,36**	95,68	**94,40**	92,12
no BERT	97,14	87,84	84,73	79,43
Martin *et al.*	*99,29*	—	*94,20*	—
Stanza	98,19	90,47	88,34	81,77
UD-Pipe 2	*99,32*	94,88	93,81	—

(d) Résultats pour UD_FRENCH-SPOKEN

Modèle	UPOS	UAS	LAS	CLAS
MBERT	*97,19*	83,93	78,13	71,05
FLAUBERT	*96,75*	86,00	80,46	73,97
no BERT	92,62	77,87	69,78	61,03
Martin *et al.*	*97,09*	—	**81,81**	—
Stanza	95,49	75,82	70,71	62,13
UD-Pipe 2	**97,23**	86,27	*81,40*	—

Les scores pour nos modèles sont obtenus en répétant la procédure d'entraînement avec trois germes aléatoires et en conservant les résultats du modèle le plus performant (en terme de LAS) sur le corpus de développement. Ces scores sont calculés par le script d'évaluation officiel de la campagne d'évaluation CoNLL 2018 [4] en considérant les métriques UPOS, UAS et LAS, ainsi que CLAS (Nivre & Fang, 2017) pour les corpus UD. Pour les métriques disponibles dans tout l'état de l'art, nous notons le meilleur résultat **en gras** et les résultats proches (moins de 0,5 points de différence) en *italiques*.

l'apprentissage des structures locales liées aux mots fonctionnels — puisque cette dernière métrique ne tiens précisément pas compte de ces dépendances.

3.2 Résultats de test

Nous présentons dans la table 2, une comparaison de nos résultats sur les jeux de données de test des principaux corpus de français à ceux de l'état de l'art — quand ils sont disponibles.

L'ensemble des modèles auxquels nous nous comparons sont également des variantes de l'algorithme d'analyse à arbre maximal couvrant introduite par Dozat & Manning (2017). Ils diffèrent essentiellement par les représentations lexicales utilisés. Notons toutefois que l'analyseur Stanza (Qi *et al.*, 2020) réalise sa propre segmentation en mots alors que l'analyseur que nous présentons utilise la segmentation proposée par le treebank. Ses résultats ne sont donc pas parfaitement comparables avec les notres et sont donnés à titre indicatif. Tous les modèles utilisent en entrée des embeddings produits par une variante de BERT à l'exception de l'analyseur de Constant *et al.* (2013) qui n'est pas un analyseur à apprentissage profond et de Stanza. UDPipe 2 (Straka *et al.*, 2019) utilise des

représentations BERT issues du modèle MBERT (Devlin *et al.*, 2019), mais contrairement à nous ces auteurs ne les ajustent pas durant l'entraînement de leur analyseur.

Les résultats obtenus par nos modèles, aussi bien pour l'étiquetage morphosyntaxique que pour l'analyse en dépendance sont généralement meilleurs que l'état de l'art, à l'exception des résultats pour UD_FRENCH-SPOKEN. Pour ce dernier corpus, nous conjecturons que les résultats moins bons que l'état de l'art que nous obtenons s'expliquent au mpins en partie par la petite taille et la plus grande irrégularité du corpus d'apprentissage, qui joue en la défaveur de nos modèles : ceux-ci présentant un plus grand nombre de paramètres que les modèles auxquels nous les comparons, leur tendance au surapprentissage est d'autant plus grande. Nous recommandons donc, pour des réutilisations de notre systèmes pour lesquelles les performances sur ce type de données seraient cruciales, une optimisation spécifique des hyperparamètres pour ce corpus afin de compenser cette baisse de performances.

Le tableau 3 donne par ailleurs des estimations des performances de ces modèles à l'inférence en terme de nombre de phrases traitées par seconde et d'occupation mémoire. On constate sans surprise que le passage sur GPU améliore grandement la vitesse, en particulier pour un modèle utilisant FlauBERT, mais que tous ces modèles restent utilisable pour des volumes de données moyens, même sur un ordinateur personnel.

TABLE 3 – Performances de nos modèles à l'inférence. Les valeurs données sont évaluées sur le corpus de test de UD_French-GSD. Les valeurs « CPU » sont mesurées sur un ordinateur portable (processeur Intel Core i7-6500U, 2.50GHz en utilisant deux cœurs) et les valeurs « GPU » sur une machine dédiée au calcul (GPU NVidia GeForce RTX 3090). Les incertitudes sont calculées sur 10 répétitions

Modèle	Vitesse CPU (phrases/s)	Vitesse GPU (phrases/s)	Mémoire vive (GiB)
FLAUBERT	4,09±0,06	20,32±0,08	5,5
no BERT	22,96±0,12	33,03±0,79	3,5

4 Conclusion

Au final, on observe que les résultats de notre analyseur sont à l'état de l'art pour le français écrit tant l'analyse syntaxique que pour l'étiquetage morphosyntaxique. On obtient ce résultat en adaptant les embeddings contextuels de BERT appris sur un modèle spécifique au français (FLAUBERT). L'apport d'autres représentations comme FastText ou les plongements de caractères semble mineur. Pour le français parlé, il reste manifestement une marge de progression qui mérite un approfondissement.

Parmi les perspectives, nous envisageons de tester cette architecture d'analyse syntaxique sur des langues moins bien dotée, comme l'ancien français, ce qui est un contexte de travail où l'utilisation de modèles de plongements contextuels est beaucoup plus complexe à mettre en oeuvre, le manque de données linguistiques disponible rendant complexe le développement de représentations lexicales telles que celles utilisées ici.

Remerciements

Ce travail bénéficié du soutien du projet ANR Profiterole (PRocessing Old French Instrumented TExts for the Representation Of Language Evolution), projet ANR-16-CE38-0010.

Références

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. doi : 10.1162/tacl_a_00051.

Bonfante G., Guillaume B. & Perrier G. (2018). *Application of Graph Rewriting to Natural Language Processing*, volume 1. ISTE Wiley.

Candito M. & Seddah D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012*, volume 2 : Association pour le Traitement Automatique des Langues.

Chu Y.-J. & Liu T.-H. (1965). On the shortest arborescence of a directed graph. *Scientia Sinica*, **14**, 1396–1400.

Constant M., Candito M. & Seddah D. (2013). The ligm-alpage architecture for the spmrl 2013 shared task : Multiword expression analysis and dependency parsing. In *Proceedings of the EMNLP Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2013) : shared task track*.

Devlin J., Chang M.-W., Lee K. & Toutanova K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota : Association for Computational Linguistics.

Dozat T. & Manning C. D. (2017). Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

Edmonds J. (1967). Optimum branchings. *Journal of Research of the National Bureau of Standards*, **71B**(4), 233. doi : 10.6028/jres.071b.032.

Gerdes K. & Kahane S. (2017). Trois schémas d'annotation syntaxique en dépendance pour un même corpus de français oral : le cas de la macrosyntaxe. In *Actes de l'atelier sur les corpus annotés du français*.

Guillaume B., de Marneffe M.-C. & Perrier G. (2019). Conversion et améliorations de corpus du français annotés en Universal Dependencies. *Traitement Automatique des Langues*, **60**(2), 71.

Hochreiter S. & Schmidhuber J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Kingma D. P. & Ba J. (2014). Adam : A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Lacheret A., Kahane S., Beliao J., Dister A., Gerdes K., Goldman J.-P., Obin N., Pietrandrea P. & Tchobanov A. (2014). Rhapsodie : a Prosodic-Syntactic Treebank for Spoken French. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation* : European Language Resource Association. hal : hal-00968959.

Le H., Vial L., Frej J., Segonne V., Coavoux M., Lecouteux B., Allauzen A., Crabbé B., Besacier L. & Schwab D. (2020). FlauBERT : Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, p. 2479–2490, Marseille, France : European Language Resources Association.

Martin L., Muller B., Ortiz Suárez P. J., Dupont Y., Romary L., de la Clergerie É., Seddah D. & Sagot B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219 : Association for Computational Linguistics.

Nivre J. & Fang C.-T. (2017). Universal Dependency Evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, p. 86–95 : Association for Computational Linguistics.

Qi P., Zhang Y., Zhang Y., Bolton J. & Manning C. D. (2020). Stanza : A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*.

Seddah D., Tsarfaty R., Kübler S., Candito M., Choi J. D., Farkas R., Foster J., Goenaga I., Gojenola Galletebeitia K., Goldberg Y., Green S., Habash N., Kuhlmann M., Maier W., Nivre J., Przepiórkowski A., Roth R., Seeker W., Versley Y., Vincze V., Woliński M., Wróblewska A. & Villemonte de la Clergerie E. (2013). Overview of the SPMRL 2013 Shared Task : A Cross-Framework Evaluation of Parsing Morphologically Rich Languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, p. 146–182 : Association for Computational Linguistics.

Straka M., Straková J. & Hajic J. (2019). Evaluating contextualized embeddings on 54 languages in POS tagging, lemmatization and dependency parsing. *CoRR*, **abs/1908.07448**.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L. & Gomez A. (2017). Attention is all you need. *Advances in neural information processing systems*, p. 5998–6008.

Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., von Platen P., Ma C., Jernite Y., Plu J., Xu C., Le Scao T., Gugger S., Drame M., Lhoest Q. & Rush A. (2020). Transformers : State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 38–45 : Association for Computational Linguistics. doi : 10.18653/v1/2020.emnlp-demos.6.

Zeman D. *et al.* (2020). Universal dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Caractérisation des relations sémantiques entre termes multi-mots fondée sur l'analogie

Yizhe Wang[1] Béatrice Daille[2] Nabil Hathout[1]
(1) CLLE, CNRS & Université Toulouse Jean Jaurès, France
(2) LN2S, CNRS & Université de Nantes, France
yizhe.wang@univ-tlse2.fr, beatrice.daille@ln2s.fr,
nabil.hathout@univ-tlse2.fr

RÉSUMÉ

La terminologie d'un domaine rend compte de la structure du domaine grâce aux relations entre ses termes. Dans cet article, nous nous intéressons à la caractérisation des relations terminologiques qui existent entre termes multi-mots (MWT) dans les espaces vectoriels distributionnels. Nous avons constitué un jeu de données composé de MWT en français du domaine de l'environnement, reliés par des relations sémantiques lexicales. Nous présentons une expérience dans laquelle ces relations sémantiques entre MWT sont caractérisées au moyen de l'analogie. Les résultats obtenus permettent d'envisager un processus automatique pour aider à la structuration des terminologies.

ABSTRACT

Semantic relations recognition between multi-word terms by means of analogy

Terminologies reflect the structure of specialised domains with the help of internally labelled relations between terms. In this article, we are interested in the recognition of conceptual relations between multi-word terms (MWTs) in vector space models. We created a dataset of semantically related MWTs of the environmental field. We present an experiment where we characterize lexical semantic relations between MWTs by means of analogy. The results show that our method can be used as a first step towards an automatic process for structuring terminology.

MOTS-CLÉS : relations terminologiques, termes multi-mots, analogie, projection sémantique.

KEYWORDS: terminological relations, multi-word terms, analogy, semantic projection.

1 Introduction

Le terme en tant qu'étiquette d'un concept d'un domaine s'inscrit dans un système linguistique spécialisé où il est mis en relation avec d'autres termes. La notion de relation joue un rôle important pour la construction des ressources terminologiques. Les dictionnaires spécialisés récents, les banques et les bases de données terminologiques, recensent les termes d'un domaine et rendre compte de l'organisation des concepts et des relations conceptuelles qui s'établissent entre eux. Les recherches actuelles sur l'identification des relations entre les termes se sont concentrées sur les termes simples (SWT) (Grabar & Hamon, 2006; Zhu *et al.*, 2016; Zhang *et al.*, 2017). Peu de travaux ont porté sur l'acquisition des relations entre MWT. Les travaux sur les relations entre MWT concernent principalement l'exploitation de la structure interne des MWT en utilisant différents types d'informations linguistiques, syntaxiques (Verspoor *et al.*, 2003) et sémantiques (Hazem & Daille, 2018).

Volume 1 : conférence principale, pages 115–124.

L'analogie est une relation proportionnelle entre deux couples de d'objets qui sont dans la même relation (Lepage & Shin-ichi, 1996; Lepage, 1998; Skousen, 2002; Claveau & L'Homme, 2005; Turney, 2008; Langlais *et al.*, 2009). Elle connaît un regain d'intérêt à la suite des travaux de Mikolov *et al.* (2013b) qui ont montré sa capacité à capter certaines relations linguistiques dans les espaces vectoriels distributionnels. Dans ce même cadre, Gladkova *et al.* (2016) ont étudié la capacité de l'analogie à capter les relations sémantiques lexicales. Plus récemment, Allen & Hospedales (2019) ont proposé une explication mathématique de l'existence des analogies dans les plongements statiques.

Dans cet article, nous nous intéressons à la caractérisation des relations entre MWT dans les espaces vectoriels distributionnels au moyen de l'analogie. Nous considérons uniquement dans ce travail les termes qui contiennent deux mots lexicaux, mais notre méthode pourrait être adaptée à des MWT de plus de deux mots. Trois groupes de relation nous intéressent : (1) ANTI qui regroupe la relation contraire et la relation contrastive ; (2) HYP qui réunit l'hyperonymie et hyponymie ; (3) QSYN, composé de la synonymie, de la quasi-synonymie et de la co-hyponymie. Notre étude s'appuie sur un jeu de données composés de couples de MWT du domaine de l'environnement, sémantiquement reliés. Ce jeu de données est construit par projection sémantique, une méthode fondée sur l'hypothèse que les MWT ont un sens compositionnel dont l'une des conséquences est que les relations sémantiques entre les SWT sont préservées dans les MWT qui les contiennent (Hamon & Nazarenko, 2001). Une annotation manuelle la préservation de ces relations a été effectuée pour les couples de MWT du jeu de données. Notre étude se distingue des travaux existants sur l'analogie entre mots ou entre termes simples (Drozd *et al.*, 2016; Liu *et al.*, 2017; Koehl *et al.*, 2020) par le fait qu'elle porte sur des relations sémantiques lexicales (hyperonymie, antonymie, synonymie) entre MWT et entre SWT comme dans *froid*:*chaud*::*air froid*:*air chaud*. Les résultats obtenus montrent que l'analogie peut être utilisée pour prédire ces relations et qu'elle fonctionne mieux pour les relations symétriques comme l'antonymie et la synonymie que pour les relations asymétriques comme l'hyperonymie et l'hyperonymie. L'analogie dans l'espace vectoriel permet réciproquement de valider les projections sémantiques des SWT.

Dans la suite de cet article, la section 2 présente un bref état de l'art sur l'acquisition des relations sémantiques au moyen de l'analogie. La section 3 présente les ressources utilisées et la construction du jeu de données. La section 4 décrit le modèle utilisé et la méthode d'évaluation. Les résultats obtenus et l'analyse des résultats sont présentés en section 5. Nous concluons notre travail et présentons les perspectives envisagées pour des travaux futurs en section 6.

2 Caractérisation des relations sémantiques par l'analogie

Les recherches actuelles sur l'analogie avec les plongements de mots se concentrent sur l'« analogie proportionnelle » du type $a : b :: c : d$ (Drozd *et al.*, 2016). Le point de départ de notre étude est (Mikolov *et al.*, 2013a) qui montre que les relations entre les mots peuvent être captées dans une large mesure par soustraction entre vecteurs distributionnels : $a - b \approx c - d$. Ainsi, les solutions d'une équation analogique $a : b :: c :?$ se trouvent parmi les vecteurs d similaires au vecteur $b - a + c$. Plus précisément, la solution de l'équation serait alors : $\mathrm{argmax}_{d \in V}(sim(d, c - a + b))$ où sim est une mesure de similarité entre vecteurs, généralement cos. Cette méthode est habituellement appelée *3cosADD*. Une autre méthode, appelée *PairDirection*, et plus fidèle à la formule initiale peut également être utilisée (Mikolov *et al.*, 2013b). La solution est alors : $\mathrm{argmax}_{d \in V}(sim(d - c, b - a))$. Levy & Goldberg (2014) ont montré que *PairDirection* est supérieure à *3cosADD* pour résoudre les

analogies syntaxiques.

Suite à (Mikolov *et al.*, 2013a), plusieurs études ont été menées sur l'évaluation quantitative de l'analogie afin de déterminer la capacité des méthodes analogiques à capter différents types de relations linguistiques (Gladkova *et al.*, 2016; Köper *et al.*, 2015). Les résultats montrent que les relations lexicales comme la synonymie sont les plus difficiles à capter parmi toutes les relations évaluées. L'analogie proportionnelle a également été utilisée par Liu *et al.* (2017) pour apprendre des plongements multi-relationnels. Dans ce travail, l'analogie est comparée à différents types de modèles sur un jeu de données composé de mots connectés par différebtes relations lexicales. Les auteurs montrent que l'analogie surpasse les autres modèles avec un rang réciproque moyen (MRR) de $0,942$ (voir section 4).

L'analogie a aussi été utilisée pour l'acquisition de relations sémantiques entre termes. Chen *et al.* (2018) l'utilisent pour identifier les relations entre les termes médicaux et Nooralahzadeh *et al.* (2018) pour trouver les antonymes et les synonymes de termes du domaine du pétrole et du gaz. Si la plupart des travaux concernent l'analogie entre les mots ou les termes simples, certaines études portent sur les relations entre termes complexes. C'est le cas de Xu *et al.* (2018) dans le domaine de la biologie qui s'intéressent à des analogies comme *carbon 14 atom*:*radioactivity*::*C4 plant*:*C4 photosynthesis*. Les travaux précédents montrent que la capacité de l'analogie à capter les relation lexicales dépend de la qualité des représentations et de la prise en compte des variations lexicales et sémantiques dans les espaces vectoriels (Hamilton *et al.*, 2016; Vu Xuan *et al.*, 2019; Saha *et al.*, 2020), de la méthode utilisée pour résoudre les équations analogiques, des relations elles-mêmes et des caractéristiques des jeux de données comme SAT (Turney *et al.*, 2003), Google (Mikolov *et al.*, 2013a) ou BATS (Gladkova *et al.*, 2016).

Notre étude se distingue de celles évoquées ci-dessus poar le fait que nous travaillons sur des termes du français du domaine de l'environnement. Comme Xu *et al.* (2018), nous travaillons aussi avec des MWT. Cependant, nos données sont plus spécifiques : (*i*) dans les analogies que nous considérons, deux SWT sont des composants de deux MWT ; (*ii*) la relation entre les MWT est identifiée en faisant l'hypothèse que s'il existe une analogie $SWT_1 : SWT_2 :: MWT_1 : MWT_2$, alors la relation entre MWT_1 et MWT_2 est la même que celle qui existe entre SWT_1 et SWT_2. La projection sémantique est une méthode basée sur l'hypothèse que le sens des MWT est compositionnel. Une conséquence de cette hypothèse est que pour deux termes multi-mots MWT_1 et MWT_2 qui ne diffèrent que par un de leurs constituants, notons C_1 celui qui apparaît dans MWT_1 et C_2 celui qui apparaît dans MWT_2 (par exemple, lorsque $MWT_1 = C_0$ prep det C_1 et $MWT_2 = C_0$ prep det C_2), la relation sémantique entre MWT_1 et MWT_2 est la même que celle qui existe entre C_1 et C_2 dans la mesure où la contribution de la partie partagée (C_0 prep det) aux sens des deux MWT est identique.

3 Matériel expérimental

Corpus. Nous avons utilisé le corpus monolingue français PANACEA Environnement (ELRA-W0065) [1]. Il se compose de 35 453 documents (environ 50 millions de mots) de différents niveaux de spécialisation. Trois opérations ont été réalisées sur le corpus : extraction du texte à partir des documents XML ; conversion en UTF-8 ; lemmatisation.

1. http://catalog.elra.info/en-us/repository/browse/ELRA-W0065/

SWT amorces. Le jeu de données est construit par projection sémantique à partir d'une liste de référence de SWT reliés par des relations sémantiques lexicales. Cette liste que nous désignerons par *RefProj* contient 831 couples de SWT nominaux et adjectivaux (116 couples ANTI ; 191 couples HYP ; 524 couples QSYN) extraits de la ressource proposée par Bernier-Colborne & Drouin (2016). *RefProj* contient des couples de termes comme : *conservation*:*protection*, *combustible*:*oil*, *flore*:*faune*.

Jeu de données. Pour générer le jeu de données [2], nous avons tout d'abord extrait du corpus PANACEA les candidats-termes qui contiennent deux mots lexicaux en utilisant l'extracteur TermSuite (Cram & Daille, 2016). La méthode de projection sémantique a ensuite été appliquée sur les couples de SWT de *RefProj* et filtrée par les candidats-termes extraits permettant ainsi d'étendre à ces derniers les relations qui existent entre les termes simples. La relation contrastive entre *flore* et *faune* peut par exemple être étendue aux termes *protection de la flore* et *protection de la faune*. Le statut de terme des candidats a été validé au moyen de trois banques terminologiques en ligne : TERMIUM Plus [3] ; Le Grand Dictionnaire [4] ; IATE [5]. À l'issue de cette validation, le jeu de données se compose de 231 couples de MWT qui se répartissent comme suit : 80 couples ANTI, 51 couples HYP et 100 couples QSYN. Une annotation manuelle de ces données a été effectuée par trois annotateurs qui ont indiqué si la relation sémantique qui existe entre les SWT est ou n'est pas préservée entre les MWT sur la base de 5 contextes extraits aléatoirement du corpus pour chaque MWT. L'accord inter-annotateurs de 0,69, ce qui reste assez fort. Une phase d'adjudication a ensuite été réalisée pour créer le jeu de données.

Les 231 couples de MWT permettent de construire 231 quadruplets formé d'un couple de SWT et d'un couple de MWT. La relations sémantique existant entre les SWT est préservée entre les MWT dans 181 de ces quadruplets (classe positive) comme pour *sec*:*humide*::*climat sec*:*climat humide* ; la classe négative est composée de 50 quadruplets comme *autoroute*:*route*::*autoroute maritime*:*route maritime* où les SWT sont dans une relation d'hyperonymie tandis que les MWT sont synonymes. La taille réduite du jeu de données s'explique par le fait que de nombreux termes extraits, comme *conservation du papillon*, sont trop spécifiques pour être présents dans les banques terminologiques que nous avons utilisées.

Le tableau 3 présente un extrait du jeu de données. *SWT1*, *SWT2* sont les deux termes simples, *MWT1* et *MWT2* les MWT qui les contiennent, *Rel* la relation entre les SWT. *Anno* indique si la relation est préservée (1) ou si elle ne l'est pas (0).

SWT1	SWT2	MWT1	MWT2	Rel	Anno
froid	chaud	air froid	air chaud	ANTI	1
piscicole	agricole	domaine piscicole	domaine agricole	ANTI	1
terre	planète	climat de la terre	climat de la planète	HYP	0
culture	agriculture	culture biologique	agriculture biologique	HYP	1
neige	glace	cristaux de neige	cristaux de glace	QSYN	0
protection	conservation	protection de l'espace	conservation de l'espace	QSYN	1

2. Le jeu de données et les ressources utilisées sont disponibles à l'adresse suivante: https://github.com/YizWang/List-of-semantically-linked-MWTs
3. https://www.btb.termiumplus.gc.ca/tpv2alpha/alpha-fra.html?lang=fra
4. http://www.granddictionnaire.com/
5. https://iate.europa.eu/

4 Expériences

Modèle. Le modèle utilisé pour identifier les analogies entre les représentations vectorielles des SWT et des MWT est construit avec FastText (Joulin *et al.*, 2017). Il se distingue par la possibilité qu'il offre de construire des modèles de plongements statiques qui combinent à la fois les mots et leurs n-grammes de caractères (Vu *et al.*, 2019), ce que Word2Vec ou Glove ne permettent pas. Cette caractéristique peut être exploitée pour construire des modèles qui contiennent en même temps des représentations pour les SWT et les MWT et dans lesquels les représentations des MWT sont indépendantes de celles des SWT qu'ils contiennent. Cela est essentiel car nous voulons comparer dans le même espace les différences entre les représentations des SWT et entre celles des MWT. Les représentations de ces derniers ne doivent donc pas être calculées compositionnellement à partir de celles de leurs constituants car les deux différences seraient alors toujours égales.

Pour créer de tels modèles, nous avons annoté dans le corpus les occurrences des MWT afin que FastText calcule leurs représentations et celles des mots qu'ils contiennent. Par exemple, les occurrences du MWT *air froid* interviennent dans le calcul des représentations de *air*, de *froid* et de *air_froid*. En pratique, nous avons remplacé dans le corpus toutes les occurrences de `air froid` par `< air froid air_froid >` et nous avons forcé FastText à ne procéder à aucun autre découpage de mots en positionnant le paramètre `-maxn` à 0. Ainsi, les représentations des MWT et celles des SWT sont générées de manière séparées dans le même espace sémantique.

Par ailleurs, la réalisation d'un MWT dans le corpus peut prendre différentes formes. Par exemple, *changement du climat* a comme variante attestée *changement climatique*. Nous avons observé que le prise en compte des variantes des MWT n'a pas d'incidence notable sur les résultats de la tâche et avons décidé de ne pas les annoter comme des occurrences des MWT.

Si les performances des plongements de mots peuvent être considérablement affectées par les hyperparamètres (Levy *et al.*, 2015), l'optimisation de l'exactitude moyenne sur un ensemble de relations hétérogènes peut ne pas être significative (Gladkova *et al.*, 2016). Par conséquent, nous n'avons pas réalisé d'optimisation des hyperparamètres. Les paramètres utilisés sont : dim=100, min_count=3, maxn=0, window_size=5, model=skipgram, epoch=20, lr=0.05 et les valeurs par défaut pour tous les autres paramètres. Par ailleurs, notre travail ne portant que sur les termes simples et multi-mots, nous avons restreint le vocabulaire du modèle aux 3 254 termes qui sont présents dans le corpus.

Évaluation. Dans cette expérience, nous nous intéressons à la classe positive, car notre objectif est de caractériser les relations entre les MWT. De ce fait, nous avons évalué notre modèle en utilisant la mesure MRR (*Mean Reciprocal Rank*) (Radev *et al.*, 2002; Chowdhury, 2010), la précision, le rappel et la F-mesure aux rangs 1, 5 et 10. Rappelons que MMR est définie comme suit :

$$MRR = \frac{1}{|W|} \sum_{i=1}^{|W|} \frac{1}{Rang_i}$$

où W représente la liste des réponses positives et $Rang_i$ correspond au rang du i-ième candidat qui appartient à W. Par ailleurs, chaque couple de MWT donne lieu à deux analogies, $SWT_1 : SWT_2 :: MWT_1 :?$ et $SWT_2 : SWT_1 :: MWT_2 :?$; les résultats présentés ci-dessous sont la moyenne des deux tests pour les relations symétriques (ANTI et QSYN). Pour les relations d'hyponymie (resp. d'hyperonymie), les résultats sont calculés séparément pour les MWT hyponymes et les MWT hyperonymes.

5 Résultats et analyses

L'expérience a été réalisée pour les deux méthodes : *3cosADD* et *PairDirection*. Les résultats obtenus étant meilleurs avec *3cosADD*, c'est cette méthode qui a été choisie pour résoudre les équations analogiques.

Résultats de la prédiction. Les résultats présentés dans la Table 1 montrent que l'analogie est assez efficace dans la prédiction des relations entre les MWT avec une MRR de 0,692. Les Tables 2 et 3 présentent les résultats obtenus pour chacune des relations. Elles montrent notamment que les relations hiérarchiques sont les plus difficiles à capter. Ces résultats sont conformes à ceux de Gladkova *et al.* (2016). Nous pouvons également observer que la MRR de l'hyperonymie est légèrement supérieure à celle de l'hyponymie, ou en d'autres termes qu'il est plus facile de prédire l'hyperonyme à partir des hyponymes que l'inverse. Par exemple, pour un quatruplet *combustible : pétrole :: gaz de combustible : gaz de pétrole*, il est plus facile de prédire *gaz de combustible* en prenant *gaz de pétrole* comme MWT inconnu au lieu de prédire *gaz de pétrole* en prenant *gaz de combustible* comme MWT inconnu.

MRR	P1	R1	F1	P5	R5	F5	P10	R10	F10
0,692	0,828	0,572	0,677	0,819	0,722	0,767	0,814	0,796	0,805

TABLE 1 – Évaluation de la capacité de l'analogie à la capture des relations sémantiques lexicales entre MWT

	ANTI	QSYN	Hyponyme	Hyperonyme
MRR	0,683	0,745	0,508	0,567

TABLE 2 – Résultats pour chaque type de relation estimés par la mesure MRR

	P1	R1	F1	P5	R5	F5	P10	R10	F10
ANTI	0,904	0,558	0,690	0,896	0,717	0,797	0,888	0,790	0,836
QSYN	0,875	0,653	0,747	0,868	0,771	0,816	0,863	0,828	0,845
Hypo	0,444	0,296	0,356	0,526	0,556	0,541	0,559	0,704	0,623
Hypero	0,526	0,370	0,435	0,552	0,593	0,571	0,559	0,704	0,623

TABLE 3 – Précision, rappel et F-mesure pour chaque type de relation sémantique lexicale

Pour la classe négative en revanche, les résultats sont insuffisants avec une F-mesure de 0,377 pour le candidat de rang 1. L'analogie dans les espaces vectoriels distributionnels n'est donc pas adaptée à l'identification des quadruplets dans lesquels la relation entre les SWT n'est pas préservée entre les MWT. Nous avons d'autre part estimé la généralité du modèle en élargissant le vocabulaire à l'ensemble de mots du corpus (120 606 mots). Le modèle prédit alors la classe positive avec un MRR de 0,6118. Ce score bien qu'inférieur à celui obtenu avec un vocabulaire limité aux termes démontre que la capacité de l'analogie à capter les relations sémantiques reste acceptable lorsque le vocabulaire est élargi.

Analyse d'erreurs. Une analyse manuelle des erreurs parmi les voisins du Top 5 a été réalisée pour identifier les quadruplets difficiles à prédire [6]. Globalement, les erreurs peuvent être due à la manière dont les plongements sont calculés ou à la méthode utilisée pour prédire les analogies. Nous avons notamment observé que l'une des causes d'erreurs principales est la polysémie des termes simples qui affecte toutes les relations sémantiques comme dans le cas de *route* qui peut désigner une infrastructure (*route terrestre*) ou une région de l'espace (*route maritime*). Le modèle s'avère également peu sensible aux variations sémantiques déterminées par le contexte comme dans le cas d'*agriculture durable* qui en discours peut être utilisé comme un hyperonyme d'*élevage durable* ou comme un antonyme de ce dernier lorsque la relation est induite par le contraste entre les plantes et les animaux.

Les difficultés de l'analogie à capter certaines relations lexicales comme l'hyperonymie explique les résultats obtenus peuvent encore être améliorés, comme l'observent Gladkova *et al.* (2016) sur un ensemble de relations plus varié. Une autre difficulté rencontrée par l'analogie est également due au fait que lorsque l'on soustrait la représentation vectorielle d'un mot M_1 à celle d'un autre mot M_2, la différence ne représente pas le sens d'un mot et qu'elle ne capte que de manière approximative la relation lexicale sémantique qui s'établie entre M_1 et M_2 (Vylomova *et al.*, 2016).

6 Conclusion et perspectives

Cet article présente une première étude de la caractérisation des relations sémantiques entre termes multi-mots dans des espaces sémantiques. Nous avons constitué un jeu de donnée composé de couples de MWT du domaine de l'environnement reliés sémantiquement. Nous avons étudié la capacité de l'analogie à identifier les relations sémantiques lexicales fondamentales entre les MWT dans les espaces vectoriels distributionnels. Ses performances sont globalement bonnes avec des différences marquées entre les relations lexicales considérées et un meilleur comportement pour les relations synmétriques. L'analyse d'erreurs fait ressortir la polysémie comme l'une des causes principales des mauvaises prédictions. La prochaine étape de ce travail est l'adaptation de cette méthode à des modèles plus sensibles au contexte comme BERT (Devlin *et al.*, 2019) pour prédire les relations sémantiques lexicales entre les MWT et la préservation des relations entre SWT qu'ils contiennent.

Références

ALLEN C. & HOSPEDALES T. (2019). Analogies explained: Towards understanding word embeddings.

BERNIER-COLBORNE G. & DROUIN P. (2016). Evaluation des modeles sémantiques distributionnels: le cas de la dérivation syntaxique. In *Proceedings the 23rd French Conference on Natural Language Processing (TALN)*, p. 125–138.

CHEN Z., HE Z., LIU X. & BIAN J. (2018). Evaluating semantic relations in neural word embeddings with biomedical and general domain knowledge bases. *BMC medical informatics and decision making*, **18**(2), 65.

CHOWDHURY G. G. (2010). *Introduction to modern information retrieval*. Facet publishing.

6. Rappelons que l'annotation du jeu de donnée est basée sur 5 contextes extraits aléatoirement tandis que la représentation vectorielle des termes est calculée à partir de l'ensemble des contextes dans lesquels ils apparaissent.

CLAVEAU V. & L'HOMME M.-C. (2005). Apprentissage par analogie pour la structuration de terminologie. Utilisation comparée de ressources endogènes et exogènes. In *Actes de la conférence terminologie et intelligence artificielle (TIA-2005)*, Rouen.

CRAM D. & DAILLE B. (2016). Terminology extraction with term variant detection. In *Proceedings of ACL-2016 System Demonstrations*, p. 13–18.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

DROZD A., GLADKOVA A. & MATSUOKA S. (2016). Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*, p. 3519–3530.

GLADKOVA A., DROZD A. & MATSUOKA S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, p. 8–15.

GRABAR N. & HAMON T. (2006). Terminology structuring through the derivational morphology. In *International Conference on Natural Language Processing (in Finland)*, p. 652–663: Springer.

HAMILTON W. L., CLARK K., LESKOVEC J. & JURAFSKY D. (2016). Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, p. 595: NIH Public Access.

HAMON T. & NAZARENKO A. (2001). Detection of synonymy links between terms: experiment and results. *Recent advances in computational terminology*, **2**, 185–208.

HAZEM A. & DAILLE B. (2018). Word embedding approach for synonym extraction of multi-word terms. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

JOULIN A., GRAVE E., BOJANOWSKI P. & MIKOLOV T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 427–431, Valencia, Spain: Association for Computational Linguistics.

KOEHL D., DAVIS C., NAIR U. & RAMACHANDRAN R. (2020). Analogy-based assessment of domain-specific word embeddings. In *2020 SoutheastCon*, p. 1–6: IEEE.

KÖPER M., SCHEIBLE C. & IM WALDE S. S. (2015). Multilingual reliability and "semantic" structure of continuous word spaces. In *Proceedings of the 11th international conference on computational semantics*, p. 40–45.

LAIGNELET M. & RIOULT F. (2009). Repérer automatiquement les segments obsolescents à l'aide d'indices sémantiques et discursifs. In A. NAZARENKO & T. POIBEAU, Éds., *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 *Actes de la 14e conférence sur le traitement automatique des langues naturelles (TALN 2007)*, p. 101–110, Toulouse.

LANGLAIS P., YVON F. & ZWEIGENBAUM P. (2009). Improvements in analogical learning: Application to translating multi-terms of the medical domain. In *Proceedings of the 12th Conférence*

of the European Chapter of the ACL (EACL 2009), p. 487–495, Athens, Greece: Association for Computational Linguistics.

LEPAGE Y. (1998). Solving analogies on words: An algorithm. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and of the 17th International Conference on Computational Linguistics*, volume 2, p. 728–735, Montréal.

LEPAGE Y. & SHIN-ICHI A. (1996). Saussurian analogy: A theoretical account and its application. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, volume 2, p. 717–722, Copenhagen.

LEVY O. & GOLDBERG Y. (2014). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, p. 171–180.

LEVY O., GOLDBERG Y. & DAGAN I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, **3**, 211–225.

LIU H., WU Y. & YANG Y. (2017). Analogical inference for multi-relational embeddings.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space.

MIKOLOV T., YIH 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*, p. 746–751.

NOORALAHZADEH F., ØVRELID L. & LØNNING J. T. (2018). Evaluation of domain-specific word embeddings using knowledge resources. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

RADEV D. R., QI H., WU H. & FAN W. (2002). Evaluating web-based question answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, p. 1153–1156, Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA).

SAHA B., LISBOA S. & GHOSH S. (2020). Understanding patient complaint characteristics using contextual clinical bert embeddings.

SERETAN V. & WEHRLI E. (2007). Collocation translation based on sentence alignment and parsing. In *Actes de la 14e conférence sur le traitement automatique des langues naturelles (TALN 2007)*, p. 401–410, Toulouse.

SKOUSEN R., Éd. (2002). *Analogical Modeling. An exemplar-based approach to language*. Volume 10 de Human Cognitive Processing. Amsterdam / Philadelphia: John Benjamins Publishing Company.

TURNEY P. D. (2008). A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of The 22nd International Conference on Computational Linguistics (COLING 2008)*, p. 905–912, Manchester.

TURNEY P. D., LITTMAN M. L., BIGHAM J. & SHNAYDER V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. *CoRR*, **cs.CL/0309035**, 482–489.

VERSPOOR C. M., JOSLYN C. & PAPCUN G. J. (2003). The gene ontology as a source of lexical semantic knowledge for a biological natural language processing application. In *SIGIR workshop on Text Analysis and Search for Bioinformatics*, p. 51–56.

VU X.-S., VU T., TRAN S. N. & JIANG L. (2019). Etnlp: a visual-aided systematic approach to select pre-trained embeddings for a downstream task.

VU XUAN S., VU T., TRAN S. & JIANG L. (2019). ETNLP: A visual-aided systematic approach to select pre-trained embeddings for a downstream task. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, p. 1285–1294, Varna, Bulgaria: INCOMA Ltd. DOI : 10.26615/978-954-452-056-4_147.

VYLOMOVA E., RIMELL L., COHN T. & BALDWIN T. (2016). Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1671–1682, Berlin, Germany: Association for Computational Linguistics. DOI : 10.18653/v1/P16-1158.

XU J., AUNG H. L. & WEERAWARDHENA S. (2018). Solving biology analogies with deep learning.

ZHANG L., LI J. & WANG C. (2017). Automatic synonym extraction using word2vec and spectral clustering. In *2017 36th Chinese Control Conference (CCC)*, p. 5629–5632: IEEE.

ZHU W., ZHANG W., LI G.-Z., HE C. & ZHANG L. (2016). A study of damp-heat syndrome classification using word2vec and tf-idf. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 1415–1420: IEEE.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Construire des ressources collaboratives pour les langues peu dotées: une modélisation orientée communauté

Elvis Mboning Tchiaze[1, 2] Ornella Wandji[1]
(1) NTeALan Research and Development, Makepe, Douala, Cameroun
(2) ERTIM, 2 Rue de Lille, Paris, France
levismboning@ntealan.org, ornella.wandji@ntealan.org

RÉSUMÉ

Les applications du traitement automatique des langues (TAL) nourrissent aujourd'hui une bonne partie des langues indo-européennes en raison des corpus linguistiques de qualité disponibles en grande quantité et variété. Les corpus de données open sources en langues africaines étant quasi inexistants, comment arrimer les avancées du TAL à ces langues peu dotées ? Dans cet article, nous examinons le problème de construction des ressources lexicographiques pour les langues peu dotées. Nous souhaitons introduire un modèle de construction des ressources lexicographiques en exploitant les compétences socio-linguistiques des communautés linguistiques locales. Au fil des sections, nous présenterons le nouveau modèle de codification des dictionnaires issue de cette modélisation orientée communauté.

ABSTRACT

Building collaborative resources for poorly endowed languages : community-oriented modeling

The applications of natural language processing (NLP) today feed a large part of Indo-European languages, with a large body of quality data available in large quantities. As open source, data corpora in African languages are almost non-existent, how can the advances in NLP be secured for these poorly endowed languages ? In this article, we address the problem of constructing lexicographic resources. We wish to introduce a model for building lexical resources by exploiting the socio-linguistic skills of local linguistic communities. Throughout the sections, we will present the new dictionary coding model resulting from this community-oriented modeling.

MOTS-CLÉS : Langues africaines, lexicographie électronique, NTeALan, modèle collaboratif, graphe, modèle basé sur la communauté.

KEYWORDS: African languages, electronic lexicography, NTeALan, collaboration model, graph, community-based model.

1 Introduction

Il y a encore quelques années de cela, les travaux de développement de ressources en langues africaines ne faisaient pas l'objet d'affluence dans le monde de la recherche. Mais de nos jours, des chercheurs, équipes de recherche, laboratoires, universités autant en Afrique qu'en Occident, et parfois en collaboration, se consacrent de plus en plus à la numérisation des langues africaines, à la conception de dictionnaires électroniques et d'autres ressources et outils du TAL dans et pour

Volume 1 : conférence principale, pages 125–133.

ces langues. L'association NTeALan Social Network [1], à travers sa jeune équipe de recherche, veut s'inscrire dans cette lignée. Depuis 2018, elle développe un modèle open source de construction collaborative des ressources lexicographiques pour les langues africaines.

En exploitant le modèle collaboratif de (Holtzblatt & Beyer, 2017), nous souhaitons dans cet article, présenter les premiers résultats des travaux de refonte des données lexicographiques de nos plateformes collaboratives construites sous un modèle orienté communauté. Il s'agit précisément de la description du nouveau modèle de codification de nos dictionnaires et l'encapsulation du modèle collaboratif vers un modèle orienté communauté inspiré des graphes. Dans les sections ci-dessous, nous présenterons d'abord les propriétés de l'ancien format de codification, ensuite celles du nouveau format et enfin la nouvelle modélisation orientée communauté.

2 XND : format initial de NTeALan

Le format XND (XML NTeALan Dictionaries) est un format de codification intermédiaire ayant une porté morpho-syntaxique, développé à partir de 2018 pour codifier et outiller les dictionnaires bilingues produits par les plateformes collaboratives de NTeALan (Mboning Tchiaze *et al.*, 2020; Mboning *et al.*, 2020).

2.1 Pourquoi créer un nouveau format de codification ?

Le choix de créer un format pour codifier nos données lexicographiques s'est fait après plusieurs observations et essais techniques exploités dans la littérature (Bosch & Pretorius, 2002, 2003; Heid, 2014; Pretorius & Bosch, 2003; Kotzé, 2005a,b; Bosch & Pretorius, 2004; Prinsloo, 2012; Bosch & Pretorius, 2011; Pretorius & Bosch, 2012; Nogwina *et al.*, 2013; Prinsloo *et al.*, 2012; Benoit & Turcan, 2006).

En effet, chaque plateforme de gestion de ressources lexicales possède son propre modèle de structuration et de présentation des données, c'est le cas des plateformes suivantes : `Kosh` (Mondaca *et al.*, 2019), `ELEXIS Dictionary Service` et `Djibiki` (Mangeot, 2006). Parmi les formats utilisés [2] pour codifier ces données, le format XML (principalement les normes TEI, CES et LMF) est aujourd'hui un choix de référence pour la structuration de données linguistiques, lexicographiques et terminographiques. Malheureusement, ces normes ne sont pas souvent adaptées pour représenter et décrire certaines particularités morpho-syntaxiques des langues africaines [3]. Pour preuve, plusieurs phénomènes linguistiques, tels que le concept de classe nominale, la notion de clics ou encore la gestion de la traduction et de la localisation des variantes dialectales de l'entrée d'article (mot vedette) ne sont pas traités explicitement, malgré tous les besoins exprimés en la matière [4].

2.2 Description du format XND : version initiale

En analysant la structure d'une langue sémi-Bantu (`yemba` parlée dans la région de l'Ouest au Cameroun), nous avons initialement décidé de définir un modèle propriétaire de structuration XML, un intermédiaire entre l'environnement interne de NTeALan et les normes externes, dont la structure

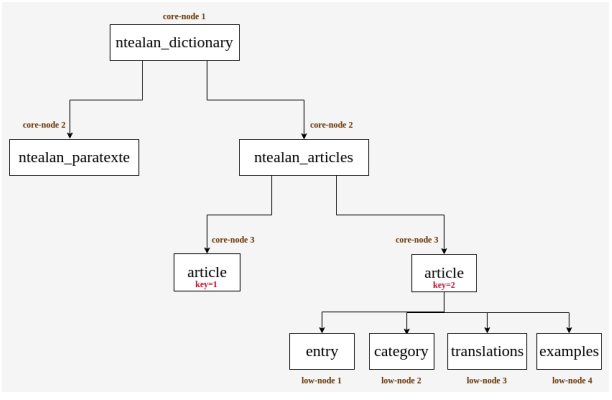
1. Elle a été initialement légalisée en 2019 au Cameroun sous le nom NTeALan. Site web officiel : https://ntealan.org
2. On peut citer les formats TEI Lex-0 (Romary & Tasovac, 2018) et Lexicog (OntoLex Lemon Lexicography du W3C), plus récents, qui sont fréquemment utilisés pour les codifier.
3. Certainement pour les mêmes raisons, aucuns des auteurs des travaux précédents sur l'xmlisation (mise au format XML) des langues africaines, n'a essayé ces formats.
4. Néanmoins, il faut noter qu'il est possible dans ces standards d'ajouter de nouvelles classes (balises et attributs) en complément des classes existantes.

s'inspirerait des 4 grandes familles de langues africaines, à savoir : la famille Afro-asiatique, la famille Niger-Kordofaniene (anciennement appelée Niger-Congo), la famille Nilo-Saharienne et la famille Khoisane. Ainsi, lors de la description de ce format, trois principes ont guidé nos choix : `représentation` (description en composants linguistiques), `simplification` (arbre et noms des balises explicites) et `extensibilité` (ouverture aux nouveaux noeuds). Cf. la figure (a) du tableau 1 :

— **Représentation** : avec ce principe, nous décrivons les données du langage au plus petit niveau morpho-syntaxique bantu, c'est-à-dire les composants du mot (variante dialectale = `préfixe + radical + suffixe`) et les composants de la phrase comme l'accord de classe (1/2, 3/4, 5/7, etc.), les préfixes d'accord. À noter que c'est l'accord de classe qui régit les structures syntaxiques dans les langues bantu et semi-bantu au sens triste du terme.
— **Simplification** : nous choisissons ici des noms de balises XML dans une langue nationale locale facilement compréhensibles par la communauté des utilisateurs. De plus, nous avons choisi d'utiliser une représentation XML linéaire, avec moins d'ascendants/descendants pour privilégier plus d'enfants du même nœud parent.
— **Extensibilité** : nous donnons aux contributeurs externes la possibilité d'étendre nos principales structures XML en ajoutant de nouveaux nœuds (enfants ou nœuds parents), en fonction de l'élément à représenter.

Plus concrètement, le nœud (balise XML) racine (*core-node*) `<ntealan_dictionary>`, est divisé en deux sous-nœuds : `<ntealan_paratexte>` et `<ntealan_articles>`. La balise `<ntealan_paratexte>` décrit les métadonnées autour de la ou des version(s) du dictionnaire (contexte de production du dictionnaire, information sur la source des données, les auteurs, l'année de création, les droits d'auteurs, etc.). Et la balise `<ntealan_articles>` décrit tous les articles du dictionnaire avec la balise enfant (`<article>`). Cf. la figure (b) du tableau 1.

(a)



(b)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ntealan_dictionary SYSTEM "DTD_NTeALan_dictionaries.dtd">

<ntealan_dictionary>

  <ntealan_paratexte> ... </ntealan_paratexte>

  <ntealan_articles> ... </ntealan_articles>

  <ntealan_tableaux_noms>
    <tabl_NP> ... </tabl_NP>
    <tabl_toponymes> ... </tabl_toponymes>
  </ntealan_tableaux_noms>

</ntealan_dictionary>
```

TABLE 1 – (a) Premier modèle de représentation initial des dictionnaires de NTeALan. (b) Première structure formelle du format XND de NTeALan

Chaque noeud article `<article>` a ses propres sous-nœuds : `<entry>` (entrée ou mot vedette de l'article constituée d'une ou plusieurs variantes dialectales `<variant>`, elle aussi est constituée des noeuds `<prefix> | <radical> | <suffix>`, partie intégrante du mot vedette), `<category>` (catégorie grammaticale en lien avec les variantes dialectales), `<classe_d_accords>` (classe d'accord `<cl_sing>` / `<cl_plur>` d'une entrée d'article de type Nom ou dans certains cas de type Adjectif), `<conjugaison>` (forme conjuguée d'une entrée d'article de type Verbe), `<translations>` (traductions associées aux variantes dialectales), `<examples>` (contextualisation des variantes dialectales).

Pour les dictionnaires ayant des entités nommées (noms de personnes et de lieux), une structuration minimale a été proposée à travers le noeud `<ntealan_tableaux_noms>`.

Il faut remarquer que le format XND, bien que spécifique aux plateformes de NTeALan, est exportable vers quelques standards existants : à ce jour il s'agit du format TEI P5 et du format LMF. Cette organisation nous permet de passer d'un format à l'autre sans perte d'informations, avec peu être une mutation structurelle et ce en fonction des outils TAL manipulés.

2.3 Limites du format XND

Après plus de deux années d'existence, les données sur la plateforme sont passées de 12 000 entrées d'article de dictionnaire à plus de 34 000 entrées. Pour être plus précis, 24 dictionnaires ont été crées avec un nombre d'articles compris entre 0 et 12 000. Les données sont accessibles via une API REST open source [5] et deux plateformes web [6].

La simplicité et la linéarité du format XND telle que voulue dès le départ a servi avec exemplarité les premières vagues d'xmlisation de dictionnaires créées par les communautés de locuteurs locaux et étudiantes. Cependant, à son état actuel, ce format ne permet pas de se projeter sur le modèle collaboratif de création des ressources synchrones ou asynchrones. Plusieurs raisons peuvent expliquer ce constat :

— Les propriétés linguistiques des 4 grandes familles des langues africaines sont sous-représentées dans le format actuel.
— La gestion des communautés de contributeurs n'a pas été codifiée dans ce format, bien qu'elle soit prise en compte dans la plateforme des dictionnaires.
— La gestion des applications externes et la standardisation ont été au coeur des enjeux de la définition du format XND.
— Le moteur de recherche appliqué à ce format n'est pas si efficace dans la recherche d'informations voulues par l'utilisateur.

Au vu de ceci, il était plus que nécessaire de faire évoluer le format XND pour mieux servir les langues africaines et leurs utilisateurs. Il devrait aussi continuer à respecter les 3 principes de base (cf. section 2.2). qui ont sous-tendu sa création.

3 Modélisation d'un nouveau format de codification des dictionnaires de NTeALan

En Afrique, d'après (Assoumou, 2017), «l'individu n'existe que pour sa communauté, cette dernière est au service de tous et de chacun dans un environnement où la solidarité, la fraternité et le respect sont des maîtres mots. Des coutumes et les traditions fondent le mode de vie des populations. Elles constituent un code social, juridique, moral, ..., hérités de la sagesse ancestrale et dont les lois font des communautés des États de droit.»

3.1 Vers un modèle collaboratif orienté communauté

Le concept de communauté n'est pas un choix anodin dans notre cas. En effet, la sociologie africaine est construite sur le modèle de la communauté, c'est-à-dire un ensemble de groupes sociaux et de

5. https://apis.ntealan.net/ntealan/dictionaries
6. https://ntealan.net/dictionaries-platform, https://ntealan.net/dictionaries

sous-groupes partageant une même langue, une même culture et un même espace géographique. Dans ces groupes, la solidarité se crée et des actions sociales émergent pour l'intérêt de tous. Ce concept montre clairement le lien culturel fort qui unit chaque citoyen à sa communauté, avant même celle de son pays (Tunde, 2012).

Notre nouveau modèle collaboratif puisera ses forces dans les sociétés communautaires africaines. Autrement dit, il exploitera les compétences socio-linguistiques locales propre à chaque communauté pour construire des ressources lexicographiques inclusives. Nous nous inspirons de l'organisation des comités de langues [7] au Cameroun pour structurer ces communautés. À chaque communauté, nous associons : les experts linguistes du comité, les enseignants/chercheurs, les locuteurs natifs et la diaspora d'une langue. Ces communautés travaillent ensemble pour créer des ressources dans leur langue au sein d'un espace collaboratif en ligne.

Cet espace collaboratif (espace de partage de compétences socio-linguistiques locales) doit répondre à la fois à toutes les exigences de ce modèle communautaire, aux exigences techniques (ergonomie, codification des données, RGPD, etc) et aux exigences scientifiques des domaines exploités (linguistique africaine, lexicographie électronique, linguistique de corpus, pédagogie/didactique, TAL). Cf. l'illustration 1.

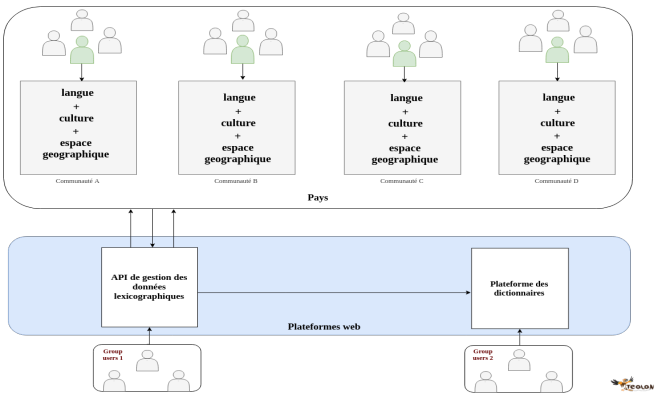


FIGURE 1 – Modèle collaboratif orienté communauté : de la constitution sociale des communautés à leur implication dans les plateformes collaboratives en ligne.

3.2 Encapsulation dans le nouveau format de codification de NTeALan

Pour ce dernier point, la représentation des données par les graphes devient une pratique assez récurrente aujourd'hui pour modéliser les connaissances ayant des liens structuraux et fonctionnels identifiables. Comme on l'a vu dans les sections précédentes, plusieurs formalismes existent sur le marché technologique, chacun ayant ses particularités, ses avantages et ses inconvénients. Comment combiner ou faire évoluer un modèle de représentation natif (le format XND initial), en un modèle de codification sémantique ouvert aux principes de la collaboration et du partage, tout en respectant le cadre socio-culturel des locuteurs natifs des langues concernées ?

Avant d'aller plus loin dans cette modélisation, nous avons d'abord voulu proposer une version améliorée de notre format XND. Il fallait donc répondre à quatre nouveaux objectifs :

7. Pour prendre l'exemple du Cameroun, chaque langue nationale a un comité de langue qui se charge de sa standardisation et de son développement. Tous ces comités de langues sont réunis autour d'une association nationale dénommée ANACLAC (Association Nationale des Comités de Langues Camerounaises).

— Normaliser l'architecture lexicographique avec l'introduction de nouveaux composants linguistiques issus de la P5 de la TEI, des entités nommées, etc.
— Standardiser les noms de balises avec un nommage en *anglais*
— Trouver les liens explicites (connexions lexicales, sémantiques, structurelles et syntaxiques) entre les balises et leurs attributs afin de créer des sous-réseaux à l'intérieur de tous les composants de l'article à la famille de langue en passant par le dictionnaire.
— Créer un système de gestion de version greffée à chaque composant afin de suivre les modifications des contributeurs en mode collaboratif.

La figure 2 montre la nouvelle représentation des composants d'un dictionnaire codifié avec le format XND. Pour mieux comprendre ce tableau, nous allons procéder par ses propriétés :

— **Les couleurs** : Les éléments en `fond jaune` représentent les balises modifiées (cf. `plur_cl, ?class_accord, source_year, +plur_cl, source_links, ?ambig_cl, +conj_form, purpose, ?plur_group, build_context, authors_version, ntealan_paratext`, etc). Les éléments en `fond vert` représentent les balises ajoutées (cf. `+date, title, ?place_table, ?definition, dictionary_name, nb`, etc.) et les éléments en `fond gris` représentent les balises existantes non modifiées (cf. `source, entry, variant, examples, institution, +article, +equivalent`, etc). Les textes écrits en `vert sous un fond blanc` représentent les attributs de balise. Entre crochet, le nom de l'attribut suivit du type numérique de sa valeur `fond gris` (cf. `[ref]:string, [etym]:string, [usg]:string`, etc) ou `string` équivaut à chaîne de caractères.
— **Les blocs** : les blocs avec une entête de couleur `noire foncée` représentent les noeuds principaux d'un point de vue structurel (cf. `articles, examples, conjugaison, ntealan_entities, entry`, etc.) et ceux avec des entêtes de couleur `rouge foncée` représentent les sous-noeuds. Certains blocs de couleurs rouges peuvent décrire le type de sa valeur (cf. `+author, +file_link`, etc), les attributs de la balise référencée (cf. `+noun, +place`) ou une sous-référence (cf. `+variant`).
— **Les symboles** : ont la même signification que les quantificateurs des expressions régulières (*****=zéro ou plusieurs fois, **+**=une ou plusieurs fois et **?**=zéro ou une fois)
— **Les connexions** : si d'une part les connexions non fléchées permettent d'établir un lien de parenté (parent / enfant) et de références (parent / ref) entre les noeuds, les connexions fléchées d'autre part permettent de décrire le contenu de l'élément fléché à partir de son parent.

Nous appelons `noeuds` tous les blocs du schéma de la figure 2 qui portent une information lexicographique définie. Les noeuds de cette nouvelle version du format XND sont organisés en deux types : le type `tag` équivaut à une balise ou un bloc (cf. `?noun_table:tag, translations:tag`, etc.) et le type `string` équivaut au contenu de la balise (cf. `author_id:string, dictionary_name:string`, etc.). Le type `string` a été généralisé sur toutes les balises pour normaliser leur contenu. Un transcodage (casting en anglais) permettrait de revenir au type initial (Exemple : passer du type 'string' `[nb]:"1"` au type 'integer' `[nb]:1`).

Un point d'honneur a été mis sur la balise du mot vedette de l'article décrivant les variantes dialectales associées [8] (`variant`). En nous appuyant sur les propriétés de la `XML TEI P5` et des travaux de (Bosch *et al.*, 2007; Bosch & Pretorius, 2011; Khoule *et al.*, 2016) nous l'avons enrichi de plusieurs

8. Nous avons choisi rester sur une description des variantes pour chaque mot vedette parce qu'elles nous permettent de mieux observer les différences linguistiques entre les sous-communautés de cette langue. Ces propriétés linguistiques seront d'un grand apport pour les taches de désambiguisation lexicale généralement constaté dans la construction des outils TAL.

nouvelles informations linguistiques afin de mieux cerner sa compréhension linguistique. Entre autres nouvelles propriétés, nous avons le `[form]` (information morphologique), `[sem]` (information sémantique), `[pron]` (prononciation), `[usg]` (cas d'usage discursif du mot), `[case]` (information sur le cas grammatical), `[syll]` (forme syllabique du mot), `[gen]` (le genre si le mot est un nom).

Il en est de même pour les noeuds `equivalent`, `example` avec les nouveaux attributs (`[syn]`, `[author]`, `[nb]`), le noeud `conj_form` avec `[mood]`, `[per]`), le noeud `cat` avec `[gen]`, `[subc]`).

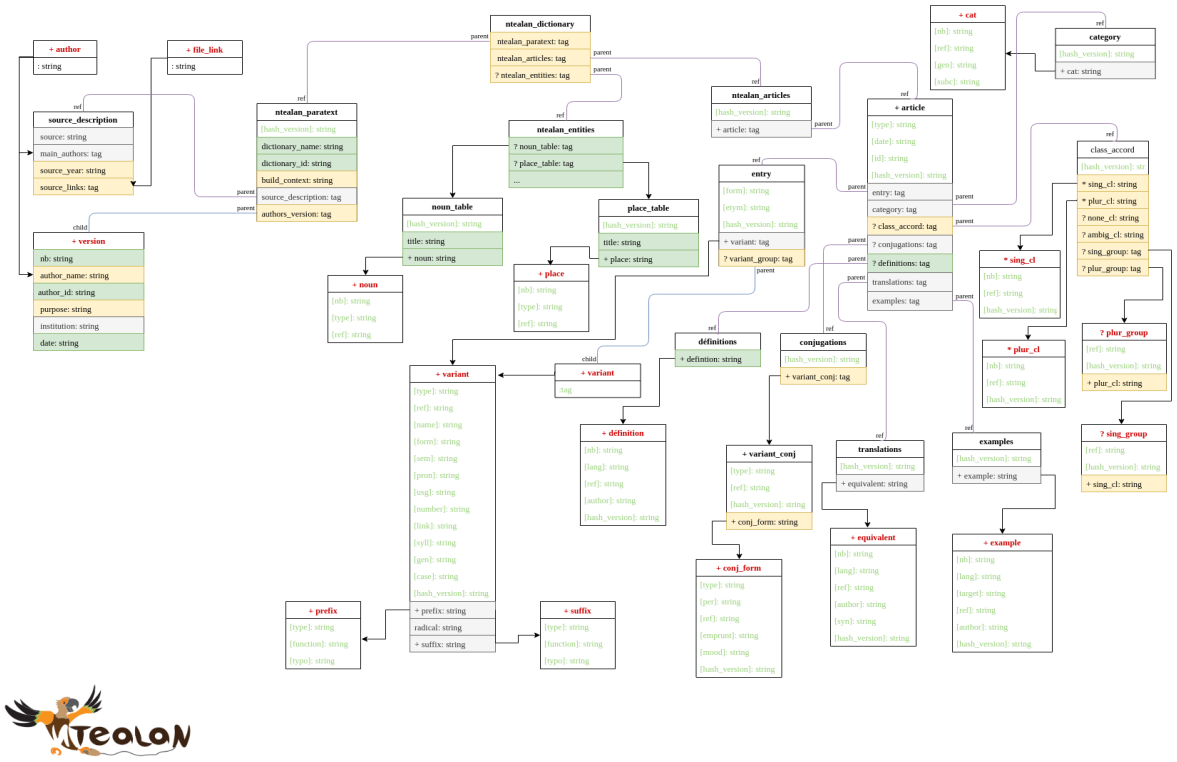


FIGURE 2 – Représentation de l'architecture améliorée du format XND et les liens entre les composants d'un dictionnaire. Regroupé en famille de langues, les dictionnaires forment un réseau.

4 Conclusion

Dans cet article, nous avons introduit un nouveau modèle collaboratif en ligne basé sur les communautés. Ce modèle orienté communauté nous permet d'associer les communautés de langues, les locuteurs natifs, chercheurs et la diaspora au sein d'un même espace de création et de partage de ressources linguistiques. Avec cette réorganisation, la nouvelle codification permettra de mieux servir les données construites par les communautés et on pourra alors envisager s'ouvrir à d'autres types de ressources autres que lexicographiques. Une mise à jour prochaine de nos plateformes permettra de déployer ce nouveau modèle afin de l'évaluer auprès de nos membres et des communautés à constituer.

Remerciements

Ce travail a été rendu possible grâce au support financier de l'équipe ERTIM de l'INALCO dans le cadre d'un projet de recherche portant sur la refonte globale du modèle collaboratif et de codification des ressources lexicographiques gérées par les plateformes collaboratives de NTeALan.

Références

ASSOUMOU J. (2017). *Culture et développement en Afrique : des perles et des pourceaux ?*, In J. ASSOUMOU & F. AMABIAMINA, Éds., *Pour une culture africaine au service du développement. Des industrie culturelles viables pour une croissance durable*, p. 14–34.

BENOIT J.-L. & TURCAN I. (2006). La TEI au service de la transmission documentaire ou de la valorisation des richesses patrimoniales : le cas difficile des dictionnaires anciens. *ANAGRAM' 2006 : Atelier sur la numérisation de l'Ecrit Ancien et des GRAndes Masses de données*.

BOSCH S. & PRETORIUS L. (2002). Finite-state computational morphology-treatment of the zulu noun. *South African computer journal*, **2002**(28), 30–38.

BOSCH S. & PRETORIUS L. (2011). Towards zulu corpus clean-up, lexicon development and corpus annotation by means of computational morphological analysis. *South African Journal of African Languages*, **31**(1), 138–158.

BOSCH S. E. & PRETORIUS L. (2003). Towards technologically enabling the indigenous languages of south africa : the central role of computational morphology. *Interactions of the Association for Computing Machinery 10 (2) (Special Issue : HCI in the developing world)*, p. 56–63.

BOSCH S. E. & PRETORIUS L. (2004). Software tools for morphological tagging of zulu corpora and lexicon development. In *LREC*.

BOSCH S. E., PRETORIUS L. & JONES J. (2007). Towards machine-readable lexicons for south african bantu languages. *Nordic Journal of African Studies*, **16**(2).

HEID U. (2014). Natural language processing techniques for improved user-friendliness of electronic dictionaries. In *of the XVI Euralex International Congress : The User in Focus*, p. 47–62.

HOLTZBLATT K. & BEYER H. (2017). *7 - Building Experience Models*. Interactive Technologies. Boston : Morgan Kaufmann. DOI : 10.1016/B978-0-12-800894-2.00007-7.

KHOULE M., MANGEOT M., NGUER E. H. M. & CISSÉ M.-T. (2016). iBaatukaay : un projet de base lexicale multilingue contributive sur le web à structure pivot pour les langues africaines notamment sénégalaises. In *Atelier Traitement Automatique des Langues Africaines TALAf 2016, conférence JEP-TALN-RECITAL 2016*, Paris, France.

KOTZÉ A. E. (2005a). Towards a morphological analyser for past tense forms in northern sotho : verb stems with final'm'and'n'. *Southern African linguistics and applied language studies*, **23**(3), 245–258.

KOTZÉ P. M. (2005b). A finite-state transducer for northern sotho deverbative nouns : the morpho-phonemic rules. *Southern African linguistics and applied language studies*, **23**(4), 393–403.

MANGEOT M. (2006). Dictionary building with the jibiki platform. In C. O. ELISA CORINO, CARLA MARELLO, Éd., *Proceedings of the 12th EURALEX International Congress*, p. 185–188, Torino, Italy : Edizioni dell'Orso.

MBONING E., BALEBA D., BASSAHAK J. M., WANDJI O. & ASSOUMOU J. (2020). NTeALan Dictionaries Platforms : An Example Of Collaboration-Based Model. In *Proceedings of the 1st International Workshop on Language Technology Platforms*, p. 66–72, Marseille, France : European Language Resources Association.

MBONING TCHIAZE E., BASSAHAK J. M., BALEBA D., WANDJI O. & ASSOUMOU J. (2020). Building Collaboration-based Resources in Endowed African Languages : Case of NTeALan Dictionaries Platform. In *Proceedings of the first workshop on Resources for African Indigenous Languages*, p. 51–56, Marseille, France : European Language Resources Association (ELRA).

MONDACA F., SCHILDKAMP P. & RAU F. (2019). Kosh : Apis for lexical data. https://github.com/cceh/kosh.

NOGWINA M., SHIBESHI Z. & MALI Z. (2013). Towards developing a stemmer for the isixhosa. *WIP. SATNAC Conference*.

PRETORIUS L. & BOSCH S. (2012). Semi-automated extraction of morphological grammars for nguni with special reference to southern ndebele. *Language Technology for Normalisation of Less-Resourced Languages*, p.73̃.

PRETORIUS L. & BOSCH S. E. (2003). Finite-state computational morphology : An analyzer prototype for zulu. *Machine Translation*, **18**(3), 195–216.

PRINSLOO D. (2012). Lexicography in non-european languages. *The Encyclopedia of Applied Linguistics*.

PRINSLOO D. J., HEID U., BOTHMA T. & FAASS G. (2012). Devices for information presentation in electronic dictionaries. *Lexikos*, **22**, 290–320.

ROMARY L. & TASOVAC T. (2018). TEI Lex-0 : A Target Format for TEI-Encoded Dictionaries and Lexical Resources. In *TEI Conference and Members' Meeting*, Tokyo, Japan.

TUNDE O. (2012). Investigating the Language Situation in Africa. In *Language and Law*, Language rights, p. 272–293. Great Clarendon street : Oxford Handbooks in Linguistics.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Contribution d'informations syntaxiques aux capacités de généralisation compositionelle des modèles `seq2seq` convolutifs

Diana Nicoleta Popa[1] William N. Havard[1]
Maximin Coavoux[1] Laurent Besacier[2] Éric Gaussier[1]
(1) Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG
(2) Naver Labs Europe
{diana.popa,william.havard,maximin.coavoux,eric.gaussier}
@univ-grenoble-alpes.fr
laurent.besacier@naverlabs.com

RÉSUMÉ

Les modèles neuronaux de type `seq2seq` manifestent d'étonnantes capacités de prédiction quand ils sont entraînés sur des données de taille suffisante. Cependant, ils échouent à généraliser de manière satisfaisante quand la tâche implique d'apprendre et de réutiliser des règles systématiques de composition et non d'apprendre simplement par imitation des exemples d'entraînement. Le jeu de données SCAN, constitué d'un ensemble de commandes en langage naturel associées à des séquences d'action, a été spécifiquement conçu pour évaluer les capacités des réseaux de neurones à apprendre ce type de généralisation compositionnelle. Dans cet article, nous nous proposons d'étudier la contribution d'informations syntaxiques sur les capacités de généralisation compositionnelle des réseaux de neurones `seq2seq` convolutifs.

ABSTRACT

Assessing the Contribution of Syntactic Information for Compositional Generalization of `seq2seq` Convolutional Networks

Classical sequence-to-sequence neural network architectures demonstrate astonishing prediction skills when they are trained on a sufficient amount of data. However, they fail to generalize when the task involves learning and reusing systematic rules rather than learning through imitation from examples. The SCAN dataset consists of a set of mapping between natural language commands and actions and was specifically introduced to assess the ability of neural networks to learn this type of compositional generalization. In this paper, we investigate to what extent the use of syntactic features help convolutional seq2seq models to better learn systematic compositionality.

MOTS-CLÉS : Compositionnalité, modèle convolutionnel seq2seq, jeux de données SCAN.

KEYWORDS: Compositionality, convolutional seq2seq model, SCAN dataset.

1 Introduction

Si les réseaux de neurones obtiennent des résultats remarquables sur différentes tâches de traitement automatique des langues (TAL), telles que la traduction automatique, les tâches de question-réponse ou la génération de texte, plusieurs études (par exemple Jia & Liang (2017) en lecture automatique,

Volume 1 : conférence principale, pages 134–141.

Input original	`jump opposite right twice and turn opposite right thrice`
ARBRE PRÉFIXE COMPLET	`(C (S (V (U jump) opposite right) twice) and (S (V turn opposite right) thrice))`
ARBRE POSTFIXE COMPLET	`((((jump U) opposite right V) twice S) and ((turn opposite right V) thrice S) C)`
ARBRE NON ÉTIQUETÉ COMPLET	`((((jump) opposite right) twice) and ((turn opposite right) thrice))`
ARBRE NON ÉTIQUETÉ PARTIEL	`(((jump opposite right) twice) and ((turn opposite right) thrice))`
CHUNKS-V	`(V jump opposite right) twice and (V turn opposite right) thrice`
Output	`RTURN RTURN JUMP RTURN RTURN JUMP RTURN RTURN RTURN RTURN RTURN RTURN`

TABLE 1: Input augmenté avec différents types d'informations syntaxiques.

ou Belinkov & Bisk (2018) en traduction automatique) montrent que leurs capacités de généralisation sont limitées. Plus récemment, il a été montré que les réseaux de neurones échouent à faire des généralisations compositionnelles simples, c'est-à-dire qu'ils ne parviennent pas à inférer le sens d'une expression complexe (*jump twice*) même en connaissant les sens respectifs de ses parties (*jump*, *twice*). Comme illustré par Dessi & Baroni (2019), les réseaux `seq2seq` convolutifs n'obtiennent qu'environ 70% d'exactitude sur le corpus de test de la tâche *jump* du jeu de données SCAN (Lake & Baroni, 2018), les GRU et LSTM étant encore plus faibles (respectivement 12.5% et 1.2%).

Ces résultats ne sont pas surprenants dans la mesure où Lake & Baroni (2018) ne donnent aucune information, à part les formes de surface, pour guider le modèle vers l'apprentissage de la compositionalité. Ces modèles apprennent essentiellement à imiter les exemples qu'ils voient lors de l'apprentissage. Ainsi, cela soulève la question de savoir si l'ajout d'informations, qui inciteraient le réseau de neurones à s'abstraire des formes de surface, amélioreraient les capacités compositionnelles du modèle. Nous proposons d'étudier cette question en analysant la contribution d'informations syntaxiques sur le jeu de données SCAN. Pour cela, un protocole expérimental est proposé avec différents types d'information syntaxiques ajoutées a priori sur les séquences d'entrée (voir Table 1) sans changer la nature de l'architecture étudiée (modèle `seq2seq` convolutif).

2 Travaux connexes

Lake & Baroni (2018) ont conçu le jeu de données SCAN pour évaluer les capacités de généralisation compositionnelle des modèles `seq2seq` classiques. Ce jeu de données comprend des commandes de navigation données en anglais simplifié et associées à des séquences d'actions. Nous donnons un exemple dans la Table 1 (lignes "*input original*" et "*output*"). Pour constituer le jeu de données, des commandes ont été générées à partir d'une grammaire hors contexte, tandis que les séquences d'actions ont été calculées à partir d'une fonction déterministe d'interprétation sémantique. Plusieurs divisions en corpus d'entraînement et d'évaluation ont été considérées sur ce jeu de données pour évaluer plusieurs types de généralisation. Dans cet article nous nous concentrons sur la tâche dite du *split* JUMP (Lake & Baroni, 2018). Dans le *split* JUMP, les seules occurrences de *jump* dans le corpus d'entraînement apparaissent de manière isolée (seules les autres actions pouvant être combinées). Dans le corpus d'évaluation, chaque exemple contient le token *jump* combiné avec d'autres tokens (*jump twice*, *jump opposite left*).

Dans l'article original de SCAN (Lake & Baroni, 2018), les réseaux récurrents (RNN, LSTM) échouent à réaliser ces généralisations. Dessi & Baroni (2019) ont évalué des modèles `seq2seq` convolutifs sur la même tâche et rapportent de bien meilleurs résultats que les RNN. Depuis ces articles fondateurs, les chercheur·es ont proposé des architectures et méthodes d'entraînement plus complexes. Par exemple, Lake (2019) propose une architecture dotée d'une mémoire (*memory-augmented*) et

entraînée par méta-apprentissage où chaque épisode d'apprentissage consiste à apprendre à combiner une nouvelle action primitive vue seulement de manière isolée.

Liu *et al.* (2020) proposent également une architecture dotée d'une mémoire et divisée en deux modules (*composer* et *solver*) . Récemment, deux modèles `seq2seq` ont obtenu de bons résultats sur le *split* JUMP. Le premier (Russin *et al.*, 2020b) présente un mécanisme d'attention basé sur deux vecteurs pour chaque mot : un vecteur contextualisé et un vecteur – dit "sémantique" – indépendant du contexte. Le second (Gordon *et al.*, 2020) présente un modèle `seq2seq` basé sur des représentations équivariantes pour certaines permutations de tokens.

Dans cet article, nous nous concentrons sur une architecture simple : un réseau `seq2seq` convolutif, et nous nous intéressons à la contribution des informations syntaxiques (données de différentes manières en input au modèle) aux capacités de généralisation compositionnelle du modèle.

3 Traits syntaxiques

Nous faisons l'hypothèse qu'une raison pour laquelle les modèles `seq2seq` échouent à apprendre la compositionalité systématique à partir de SCAN est que ces systèmes ne modélisent pas la structure syntaxique des énoncés. Pour tester cette hypothèse, nous proposons d'augmenter les exemples d'input de SCAN avec des informations syntaxiques explicites et d'évaluer si cela permet d'améliorer les capacités de compositionalité systématique des modèles. [1]

Pour cela, nous commençons par faire l'analyse syntaxique de chaque input à l'aide d'un analyseur CKY standard [2] et de la grammaire non ambiguë fournie en annexe de l'article SCAN (Lake & Baroni, 2018). Ensuite, nous augmentons les phrases d'input pour qu'elles aient l'une des formes suivantes :

i Arbre complet étiqueté en notation préfixe (ARBRE PRÉFIXE COMPLET) ;
ii Arbre complet étiqueté en notation postfixe (ARBRE POSTFIXE COMPLET) ;
iii Arbre complet non étiqueté (ARBRE NON ÉTIQUETÉ COMPLET) ;
iv Arbre partiel non étiqueté (ARBRE NON ÉTIQUETÉ PARTIEL), obtenu en retirant les parenthèses correspondant aux règles de grammaire unaire [3] ;
v Analyse syntaxique superficielle (syntagmes étiquetés V, CHUNKS-V).

Nous présentons la forme de l'entrée pour chaque cas dans la Table 1. Les deux premiers types d'input (i-ii) codent le maximum d'information syntaxique : parenthésage complet et étiquetage des syntagmes par une des étiquettes suivantes :

— U représente une action primitive (`walk, look, jump, run`) ;
— D représente un syntagme complexe comprenant un changement de direction (`turn, left, right`) ;
— V est utilisé pour des syntagmes ayant potentiellement un modifieur tel que `opposite` ou `around` ;
— S représente une proposition complète (potentiellement plusieurs actions primitives et leurs modifieurs) ;
— C représente des coordinations.

1. Les données augmentées sont librement disponibles : `https://github.com/mcoavoux/SCAN_syntax`.
2. Nous utilisons l'implémentation de la bibliothèque NLTK (Bird *et al.*, 2009).
3. Les tokens `walk`, `look`, `run` et `jump` ne peuvent être générés que par des règles unaires. Tous les non-terminaux (sauf l'axiome C) peuvent aussi être générés par des règles unaires.

Les trois derniers types d'input (ARBRE NON ÉTIQUETÉ COMPLET, ARBRE NON ÉTIQUETÉ PARTIEL et CHUNKS-V) permettent d'isoler les contributions des informations de structures et d'étiquetage : Le type d'input (iii) contient toute la structure syntaxique mais pas d'étiquette. Le type (iv) représente l'arbre non étiqueté et sans le parenthésage correspondant aux règles unaires. Par exemple, `jump` (Table 1) est au même niveau que `opposite` et `right` pour le protocole ARBRE NON ÉTIQUETÉ PARTIEL. Enfin, le type (v) représente un unique type de syntagme (V). Nous considérons les parenthèses ouvrantes et fermantes, ainsi que les étiquettes des syntagmes comme des tokens. Ces inputs enrichis permettent ainsi d'entraîner une même architecture en faisant varier la quantité d'information syntaxique fournie afin d'en étudier la pertinence et l'utilisation par un réseau `seq2seq` convolutif.

Le vocabulaire des versions enrichies en syntaxe que nous avons construites comprend 20 symboles au plus : les 13 symboles initiaux (setting ORIGINAL), ainsi que les parenthèses et éventuellement les non-terminaux.

4 Expériences

Protocole expérimental Comme Dessi & Baroni (2019), nous utilisons un modèle de type encodeur-décodeur convolutif (Gehring *et al.*, 2017), via l'implémentation de `fairseq` (Ott *et al.*, 2019) v.0.9.0, [4] et PyTorch 1.5.0. Ce choix est motivé par les conclusions de Dessi & Baroni (2019) qui montrent des résultats empiriques nettement superieurs par rapport aux LSTM. [5] À l'instar de travaux précédents (Loula *et al.*, 2018; Lake & Baroni, 2018; Dessi & Baroni, 2019) et pour assurer une comparabilité des résultats, nous dupliquons l'ensemble d'exemples d'entraînement original pour obtenir 100 000 exemples. Cependant, nous utilisons une méthode légèrement différente [6] : nous commençons par échantillonner aléatoirement 10% des données d'entraînement pour les utiliser comme ensemble de validation. Parmi les 90% restant, nous obtenons le jeu de données augmenté en dupliquant chaque commande pour qu'elle apparaîsse 7 fois. Pour le split JUMP, cela nous donne 92 421 exemples d'entraînement et 1 467 exemples pour la validation. Nous évaluons sur le même ensemble de test que les travaux précédents (7 706 exemples).

Pour calibrer les hyperparamètres, nous faisons varier le pas d'apprentissage (0.1, 0.01), la taille des batches (50, 100, 200, 500, 1000), le nombre de couches cachées (de 5 à 10), la taille des vecteurs de mots (128, 256, 512), la probabilité de *dropout* (0, 0.25, 0.5) et la taille du noyau de convolution (3, 4, 5). Ces plages de valeurs sont similaires à celles explorées dans des travaux précédents. Nous entraînons chaque configuration du modèle pendant une époque (tel que fait dans (Lake & Baroni, 2018)) et répliquons le procédé trois fois avec des graines aléatoires distinctes.

Effet des Informations Syntaxiques Nous présentons dans la Table 2 les résutats obtenus avec les meilleurs modèles en moyennant (ou non) sur 3 graines, c'est-à-dire en moyennant sur trois entrainements de la même architecture dont le seul paramètre de variation est l'initialisation des poids (*via* la graine aléatoire). Tandis que les scores non moyennés représentent une limite haute de scores qui peuvent effectivement être obtenus avec une architecture donnée, la moyenne des scores des 3 graines représente plus fidèlement un niveau moyen de performances d'un même architecture. Dans

4. https://github.com/pytorch/fairseq

5. Ils attribuent cette supériorité des CNN aux biais d'induction des *kernels* de convolutions qui traitent le texte par fenêtres de mots et favorisent l'apprentissage de patrons.

6. L'article original de SCAN (Lake & Baroni, 2018) échantillonne avec remplacement 100k exemples parmi les 14 670 exemples distincts du corpus d'entraînement.

	Exactitude tous modèles				Exactitude tous modèles, moyenné sur 3 graines aléatoires			
	Top 1	Top 5	Top 10	$\Delta(\%)$	Top 1	Top 5	Top 10	$\Delta(\%)$
ORIGINAL	84.88	81.89($\pm$2.01)	79.44($\pm$2.96)	-	68.39($\pm$4.3)	67.19($\pm$0.69)	66.04($\pm$1.33)	-
CHUNKS-V	73.05	71.10($\pm$1.21)	69.31($\pm$2.01)	-13.93	60.92($\pm$5.27)	59.29($\pm$0.87)	57.12($\pm$2.66)	-10.92
ARBRE NON ÉTIQUETÉ PARTIEL	82.98	77.93($\pm$3.53)	74.53($\pm$4.3)	-2.23	65.06($\pm$4.66)	63.92($\pm$1.05)	61.46($\pm$2.72)	-4.86
ARBRE NON ÉTIQUETÉ COMPLET	95.92	92.04($\pm$3.38)	88.37($\pm$4.41)	$+13.01$	80.62($\pm$10.86)	76.99($\pm$2.75)	73.47($\pm$4.04)	$+17.88$
ARBRE PRÉFIXE COMPLET	97.71	93.66($\pm$2.37)	91.54($\pm$2.78)	$+15.11$	**87.29**($\pm$4.66)	**84.56**($\pm$2.22)	**81.21**($\pm$3.45)	$+27.64$
ARBRE POSTFIXE COMPLET	**98.22**	**95.99**($\pm$1.75)	**93.05**($\pm$3.24)	$+15.72$	86.54($\pm$14.45)	81.27($\pm$2.65)	79.75($\pm$2.41)	$+26.54$
Gordon *et al.* (2020)	91.0 $\pm$ 27.4 (moyenne de 25 entraînements, médiane : 98.5)							
Russin *et al.* (2020b)	**99.1** $\pm$ 0.04 (moyenne de 5 entraînements, entraînement sur 200k itérations)							

TABLE 2: Exactitude ($\pm$ écart type) sur le corpus de test du split JUMP. ORIGINAL : données originales de Lake & Baroni (2018), voir la section 3 pour la description des autres configurations. *Top-1 tous modèles* : meilleure exactitude pour la meilleure graine aléatoire d'une configuration. *Top-1 tous modèles moyenné sur 3 graines aléatoires* : meilleure configuration où le score d'une configuration est calculée comme une moyenne des scores données par 3 graines aléatoires. Top *k* scores : score moyenné des *k* meilleurs modèles. La colonne Δ indique la variation relative entre le score *Top-1* d'un type d'input et le *Top-1* de l'input baseline.

Paramètres	ORIGINAL	CHUNKS-V	ARBRE NON ÉTIQUETÉ PARTIEL	ARBRE NON ÉTIQUETÉ COMPLET	ARBRE PRÉFIXE COMPLET	ARBRE POSTFIXE COMPLET
Tous modèles						
lr	0.01	0.01	0.01	0.01	0.01	0.01
bsz	50	100	100	50	50	50
layer	10	9	7	7	5	5
dim	512	512	512	512	512	512
kernel	3	3	5	3	5	3
dp	0.25	0.25	0.25	0.25	0.25	0.25
seed	2	3	3	3	2	1
Tous modèles, moyenné sur 3 graines aléatoires						
lr	0.1	0.1	0.01	0.01	0.01	0.01
bsz	500	500	100	50	50	50
layer	10	10	7	7	9	7
dim	256	128	512	512	512	512
kernel	5	5	3	3	3	3
dp	0.25	0.25	0.25	0.25	0.25	0.25

TABLE 3: Hyperparamètres des meilleures configurations sur le split JUMP. lr : pas d'apprentissage, bsz : taille des batches, layer : nombre de couches cachées, dim : dimension des vecteurs de mots, dp : probabilité de *dropout*.

les deux cas, nous présentons les scores pour les modèles qui sont dans le Top 1, Top 5 et Top 10 [7] ainsi que les écarts-types correspondants. Les différences de performances relatives par rapport aux modèles Top 1 entrainés sur ORIGINAL sont aussi présentées pour chacun des modèles entraînés avec différent niveaux d'annotations syntaxiques. Les hyperparamètres utilisés pour obtenir ces différents résultats sont présentés dans la Table 3.

Nous pouvons observer une amélioration de près de 16% sur les meilleurs scores et de près de 28% sur les scores moyennés lorsque nous utilisons des données syntaxiquement annotées. Que l'information soit présentée de manière préfixée ou suffixée (ARBRE PRÉFIXE COMPLET ou ARBRE POSTFIXE COMPLET) n'impacte pas significativement les résultats. On constate cependant une large différence dans les résultats obtenus avec ARBRE NON ÉTIQUETÉ COMPLET et ARBRE NON ÉTIQUETÉ PARTIEL, le premier étant meilleur que le second. Dans le premier cas, les règles unaires sont explicitement parenthésées alors que dans le second cas elles ne le sont pas. Parenthèser les règles unaires, qui sont utilisées pour générer l'ensemble des commandes primitives qui se comportent de la même manière (`run`, `jump`, `look`, `walk`) permet donc aux modèles de mieux généraliser.

7. 1 620 architectures en tout pour 4 860 modèles au total.

(a) Ts. les modèles FULL UNL. TREE

(b) Ts. les modèles (moy.) FULL UNL. TREE

(c) Ts. les modèles ARBRE POSTFIXE COMPLET

(d) Ts. les modèles (moy.) ARBRE POSTFIXE COMPLET

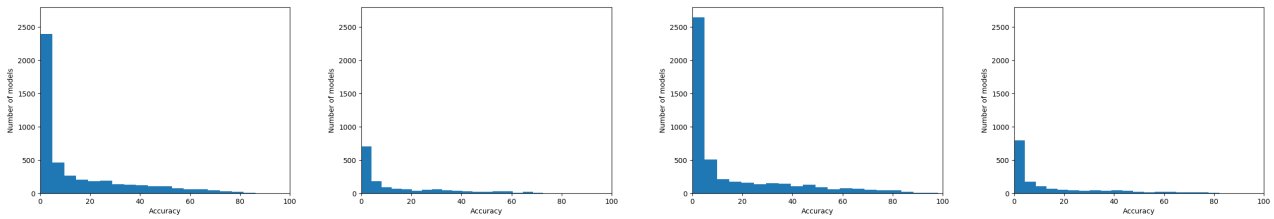


FIGURE 1: Histogrammes de l'exactitude sur les données de test de tous les modèles entrainées sur ARBRE NON ÉTIQUETÉ COMPLET sans faire la moyenne sur les différentes graines (a), en faisant la moyenne sur 3 graines (b) et pour ARBRE POSTFIXE COMPLET sans faire la moyenne sur les différentes graines (c), en faisant la moyenne sur 3 graines (d).

Ces résultats semblent indiquer que la tâche la plus ardue pour les modèles est d'isoler correctement les commandes primitives. En effet, les résultats chutent lorsque cette information n'est pas présente. En isolant un petit ensemble de commandes, le parenthésage rend l'analyse lexicale des tokens plus facile. Pour terminer, les résultats obtenus dans la configuration CHUNKS-V sont surprenamment inférieurs à ceux obtenus sur les données ORIGINAL, quand bien même une information sur la structure syntaxique, bien que minimale, soit présente. Cela peut s'expliquer par le fait que les modèles doivent traiter des séquences plus longues sans nécessairement avoir un gain proportionnel d'information pertinente en entrée.

Variation de graine. Nos résultats sont en accord avec les résultats de précédentes recherches. En effet, nous observons de fortes variations de performances lorsqu'une même architecture neuronale est entraînée avec différentes graines, comme le montrent les écarts types des meilleurs modèles (Top 1) de la Table 2. Nous présentons plus précisément comment cette variation se répartit dans différentes configurations syntaxiques dans la Figure 1. Cette figure montre la dispersion des résultats pour toutes les configurations d'hyperparamètres (en moyennant ou non sur 3 graines), pour ARBRE NON ÉTIQUETÉ COMPLET et ARBRE POSTFIXE COMPLET. Comme on peut le constater, la proportion de mauvais modèles (exactitude proche de 0) est très haute, aussi bien pour ARBRE NON ÉTIQUETÉ COMPLET que pour ARBRE POSTFIXE COMPLET. De plus, la proportion de modèles avec une exactitude supérieure à 50 décroît lorsque les résultats sont moyennés sur 3 graines. Cela explique la variance relativement haute que l'on peut observer dans la Table 2.

Perspectives architecturales. En analysant les configurations neuronales qui obtiennent les meilleurs scores, nous observons que celles-ci sont moins profondes lorsque de l'information syntaxique est présente (seulement besoin de 7 et 5 couches pour la meilleure graine et le meilleur modèle dans la configuration ARBRE POSTFIXE COMPLET), que lorsque cette information n'est pas présente (besoin de 9 et 10 couches pour obtenir les meilleurs performances pour ORIGINAL et pour les modèles entrainés sur des arbres syntatixques non-étiquettés). Ainsi, il semblerait que l'ajout d'informations syntaxiques permette d'entraîner des modèles moins profonds.

C'est d'autant plus remarquable que les séquences à traiter sont plus longues lorsqu'elles ont été annotées syntaxiquement (d'un facteur 2 pour CHUNKS-V et d'un facteur 4 pour les arbres complets) et que de longues séquences peuvent être fastidieuses à traiter pour des CNN avec un nombre limité de couches. Les meilleures performances que nous observons peuvent être dues au fait que les modèles

n'ont pas à inférer l'arbre syntaxique des phrases par eux-même et peuvent ainsi se concentrer sur l'apprentissage de la manière dont les différentes configurations syntaxiques affectent les sorties.

5 Conclusion

Nous avons étudié de quelle manière fournir des informations syntaxiques peut améliorer les capacités de généralisation compositionnelle de modèles séquence à séquence convolutifs. Pour ce faire, nous avons annoté les données du corpus SCAN avec différents niveaux d'informations syntaxiques, et avons évalué l'effet de telles annotations sur des architectures séquence à séquence convolutives standards. Nous avons observé que la structure brute de l'arbre syntaxique est l'information la plus importante pour permettre aux modèles neuronaux d'apprendre la compositionalité, l'étiquetage des syntagmes n'apportant que des gains de performances relatifs. De plus, le calibrage des hyper-paramètres trouve fréquement des modèles moins profonds lorsque ceux-ci traitent des données syntaxiquement annotées que lorsqu'ils traitent les données non-annotées. Cela suggère que les informations supplémentaires réduisent le besoin d'entrainer des modèles plus profonds, malgré l'augmentation de la longueur de la séquence d'entrée à traiter.

Dans de futurs travaux, nous souhaiterions travailler sur des partitions de données de SCAN plus difficiles comme la partition AROUND RIGHT (Loula *et al.*, 2018). Nous souhaiterions également étudier les capacités de généralisation d'autres architectures séquence à séquence, telles que celles proposées par Russin *et al.* (2020a,b) et Gordon *et al.* (2020), ainsi que sur des architectures basées sur Transformers (Vaswani *et al.*, 2017), telles que récemment étudiées sur un autre corpus par Hupkes *et al.* (2020).

6 Remerciements

Ce travail a été partiellement soutenu par l'Institut Multidisciplinaire en Intelligence Artificielle MIAI@Grenoble Alpes (ANR-19-P3IA-0003).

Références

BELINKOV Y. & BISK Y. (2018). Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

BIRD S., KLEIN E. & LOPER E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st édition.

DESSI R. & BARONI M. (2019). CNNs found to jump around more skillfully than RNNs : Compositional generalization in seq2seq convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3919–3923.

GEHRING J., AULI M., GRANGIER D. & DAUPHIN Y. (2017). A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 123–135.

GORDON J., LOPEZ-PAZ D., BARONI M. & BOUCHACOURT D. (2020). Permutation equivariant models for compositional generalization in language. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

HUPKES D., DANKERS V., MUL M. & BRUNI E. (2020). Compositionality decomposed : How do neural networks generalise ? *Journal of Artificial Intelligence Research*, **67**, 757–795. DOI : 10.1613/jair.1.11674.

JIA R. & LIANG P. (2017). Adversarial examples for evaluating reading comprehension systems. In M. PALMER, R. HWA & S. RIEDEL, Éds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, p. 2021–2031 : Association for Computational Linguistics.

LAKE B. M. (2019). Compositional generalization through meta sequence-to-sequence learning. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D' ALCHÉ-BUC, E. FOX & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 32*, p. 9791–9801. Curran Associates, Inc.

LAKE B. M. & BARONI M. (2018). Generalization without systematicity : On the compositional skills of sequence-to-sequence recurrent networks. In J. G. DY & A. KRAUSE, Éds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 de *Proceedings of Machine Learning Research*, p. 2879–2888 : PMLR.

LIU Q., AN S., LOU J.-G., CHEN B., LIN Z., GAO Y., ZHOU B., ZHENG N. & ZHANG D. (2020). Compositional generalization by learning analytical expressions. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. F. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems*, volume 33, p. 11416–11427 : Curran Associates, Inc.

LOULA J., BARONI M. & LAKE B. (2018). Rearranging the familiar : Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 108–114, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5413.

OTT M., EDUNOV S., BAEVSKI A., FAN A., GROSS S., NG N., GRANGIER D. & AULI M. (2019). fairseq : A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019 : Demonstrations*.

RUSSIN J., JO J., O'REILLY R. & BENGIO Y. (2020a). Compositional generalization by factorizing alignment and translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : Student Research Workshop*, p. 313–327, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-srw.42.

RUSSIN J., JO J., O'REILLY R. C. & BENGIO Y. (2020b). Systematicity in a recurrent neural network by factorizing syntax and semantics. In *Proceedings of the 42th Annual Meeting of the Cognitive Science Society, CogSci 2020, virtual, July 29 - August 1, 2020* : cognitivesciencesociety.org.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 30*, p. 5998–6008 : Curran Associates, Inc.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Defining And Detecting Inconsistent System Behavior in Task-oriented Dialogues

Léon-Paul Schaub[1, 2] Vojtěch Hudeček[3] Daniel Štancl[3]
Ondřej Dušek[3] Patrick Paroubek[1]
[1]Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France
[2]Akio, 43 rue de Dunkerque, 75010 Paris, France
[3]Charles University, Faculty of Mathematics and Physics, Malostranské náměstí 25, 118 00 Prague, Czechia
schaub@limsi.fr, hudecek@ufal.mff.cuni.cz, stancl@ufal.mff.cuni.cz,
odusek@ufal.mff.cuni.cz, pap@limsi.fr

RÉSUMÉ

Définition et détection des incohérences du système dans les dialogues orientés tâche.
Nous présentons des expériences sur la détection automatique des comportements incohérents des systèmes de dialogues orientés tâche à partir du contexte. Nous enrichissons les données bAbI/DSTC2 (Bordes *et al.*, 2017) avec une annotation automatique des incohérences de dialogue, et nous démontrons que les incohérences sont en corrélation avec les dialogues ratés. Nous supposons que l'utilisation d'un historique de dialogue limité et la prédiction du prochain tour de l'utilisateur peuvent améliorer la classification des incohérences. Si les deux hypothèses sont confirmées pour un modèle de dialogue basé sur les réseaux de mémoire, elles ne le sont pas pour un entraînement basé sur le modèle de langage GPT-2, qui bénéficie le plus de l'utilisation de l'historique complet du dialogue et obtient un score de précision de 0,99.

ABSTRACT

We present experiments on automatically detecting inconsistent behavior of task-oriented dialogue systems from the context. We enrich the bAbI/DSTC2 data (Bordes *et al.*, 2017) with automatic annotation of dialogue inconsistencies, and we demonstrate that inconsistencies correlate with failed dialogues. We hypothesize that using a limited dialogue history and predicting the next user turn can improve inconsistency classification. While both hypotheses are confirmed for a memory-networks-based dialogue model, it does not hold for a training based on the GPT-2 language model, which benefits most from using full dialogue history and achieves a 0.99 accuracy score.

MOTS-CLÉS : système de dialogue orienté-tâche, incohérences, modèle utilisateur, apprentissage automatique.

KEYWORDS: task-oriented dialogue systems, inconsistency, user model, machine learning.

1 Introduction

Compared to traditional pipeline architectures, the recent end-to-end neural-network-based dialogue systems have a simpler design and less space for error accumulation, but suffer from less control over the training, reduced explainability and a need for large amounts of training data and computing power, not to mention the difficulty to incorporate external knowledge bases. To address these problems, Madotto *et al.* (2018) developed Mem2Seq, an end-to-end architecture based on memory

Volume 1 : conférence principale, pages 142–152.

networks (Weston *et al.*, 2015; Sukhbaatar *et al.*, 2015) for learning from an external database with a relatively small model. Chen *et al.* (2019) then built WMM2Seq, a Mem2Seq-based dialogue system inspired by cognitive science research, whose architecture is composed of two memory networks, one learning from the dialogue history and the other from a knowledge base. Other neural-based works build on two-stage decoding within the same network (Hosseini-Asl *et al.*, 2020; Ham *et al.*, 2020). However, neither architecture solves the problem of system inconsistencies inherent in any dialogue generation task. Indeed, during a human-machine conversation, it is not uncommon to observe the machine saying something unexpected or inconsistent (Litman *et al.*, 2006; Engelbrecht & Möller, 2010). A detection and correction of these inconsistencies is difficult, but would constitute an important improvement since it would allow the system to correct itself (Zhang *et al.*, 2019), bringing us one step closer to a lifelong learning architecture (Veron, 2019; Hancock *et al.*, 2019). In this paper, we make the following contributions: (1) we enrich a task-oriented dialogue dataset with inconsistencies annotation, (2) we show that dialogue inconsistencies correlate with failures of the respective dialogues and (3) we perform a series of experiments to train and evaluate inconsistency classification models based on history and user modeling.

2 Related Works

In machine translation, Ma *et al.* (2019) showed that an incremental/simultaneous translation model can get faster by anticipating sequences, with results close to full sentence translation. In dialogue, Shang *et al.* (2020) reached state-of-the-art results for dialogue act classification by labeling speaker change in dialogue turns during learning, which shows the importance of speaker roles in the conversation. Auguste *et al.* (2019) take this idea one step further by learning to classify the dialogue act of the current and the next dialogue turn, with comparable results. Finally, Lin *et al.* (2020) create a system called "Imagine then Arbitrate" (ITA) to learn when to answer and when to listen, by imagining what the user will say to anticipate possible system errors.

Regarding system error analysis, Whitney *et al.* (2017) model with a POMDP (Sammut & Webb, 2010) the uncertainty of a dialogue agent when answering a user question to improve the answer accuracy. Welleck *et al.* (2019) use a natural language inference model to improve a system's consistency in a dialogue, Li *et al.* (2020) then integrate consistency into the system training signal. Dziri *et al.* (2019) apply a similar inference-based approach for dialogue system evaluation. During the DSTC6 shared task (Hori *et al.*, 2019), inconsistency detection for non-task-oriented dialogues was one of the problems investigated; however, the inconsistencies found remain quite specific to this type of dialogue. Gao *et al.* (2019) show that when a conversation exceeds a certain number of dialogue turns, end-to-end systems see their performance decrease, which they attribute to the the conversation history becoming noisy if it is too large. To our knowledge, there has not been a system that predicts the next user's turn and filters dialogue history to anticipate system inconsistencies.

3 Inconsistency Classification

We need to distinguish between understanding or decision errors in human-human dialogues, and bot-specific inconsistencies in a human-machine dialogue. Indeed, during a task-oriented conversation between two humans, errors or problems lead to almost systematic co-corrections between the two

interactors. Self-initiated and self-repaired or hetero-initiated co-corrections are applied whenever an error occurs (Chernyshova, 2018). However, these errors are corrected thanks to inference, i.e. the explicit errors of a speaker cause the listener to interpret and infer what the first speaker really meant. In the same way, the speaker who made the error infers what the listener could have understood and inferred, in order to self-correct (Deppermann, 2018; Fernandez *et al.*, 2006). In human-machine interaction, this inference is impeded by human knowing that they are talking to a machine with an a priori limited capacity for inference and correction. Therefore, when the user or the machine commits human-like errors, this limitation inevitably lowers the confidence and patience on the user part. The main focus in this paper is, however, on *inconsistencies* – system errors that are not human-like but endemic to the system's learning process. Inconsistencies may often cause the user to end the dialogue unsuccessfully (cf. our experiment in Section 4.2). For the machine to acquire some co-correction capability, i.e. to be able to detect inconsistencies in its own responses, it must have some representation of the user interpreting them. The DSTC2 dialogues are between a real system and users (Henderson *et al.*, 2014a), and thus include many system inconsistencies, especially based on speech recognition errors. A lot of these inconsistencies are recoverable , so the whole dialogue is not jeopardized. Based on an empiric analysis of the corpus, we list the most frequent types of inconsistencies found in human-computer interaction, with an example for each of them (see Table 1 for statistics):

1. **Intent misunderstood** (*Ex: U -> in which zone is it located? S -> Eraina is a great restaurant.*) Here, the system did not predict the correct user intent (*request*) and slot (*area*).
2. **Bad entity** (*Ex: U -> Mediterranean. S -> There is no Jamaican restaurant here.*) We distinguish this case from the previous one because in this case, the system actually predicted the correct intent and slot, but it then generated an incorrect output.
3. **Repetition** (*Ex: U -> gastropub. S -> What type of restaurant are you looking for? U -> Gastropub. S -> Ok let me look.*) The system forced the user to repeat the same utterance.
4. **"Bye" misunderstood** (*Ex: U -> goodbye. S -> The restaurant Paris serves French cuisine.*) This is a particular type of 1. where the system does not understand that the user wants to end the conversation.
5. **User insults** (*Ex: U -> i dont give a fuck if hk_fusion serves chinese food im looking for thai food. S -> am sorry but there is no fusion restaurant that matches your request.*) In addition to providing a clue on intent misunderstanding, this category also provides feedback about the user feeling, which is useful when considering commercial deployment (Kumar, 2010).
6. **Bad API call** (*Ex: U -> in the south. S -> api call = Kitchen, ANY_location, expensive*). As the API call is explicit, we can verify easily if it is consistent with the information given by the user.
7. **Incompleteness** (*Ex: U -> phone and address. S -> Phone is* ***) The system only responds to some of the slots requested by the user.
8. **Negation error** (*Ex: U -> no french food. S ->There are restaurants serving french food . What area do you want?*) The system is unable to process a negation in the user's request.

A lot of the above inconsistencies can be detected using simple rules (e.g., detecting that the system utterance repeats twice in a row). More elaborate rules may concern entity type mismatch between the user and system utterances, etc. The main source of information for all the rules is the dialogue history. However, a conversation history that is too large can become a source of noise for the system (Madotto *et al.*, 2018). According to several studies (Wolters *et al.*, 2009; Lai *et al.*, 2020), the longer a dialogue history is, the more difficult it becomes for the system to retrieve the most relevant

inconsistency type								total inconsistent	correct turns	total turns
1	2	3	4	5	6	7	8			
783	245	1,360	275	11	242	780	64	3,760	26,179	29,939

Table 1 – Number of inconsistencies of various types (see Section 3 for explanation of the individual types) in the bAbI corpus.

information, especially if the slots change during the dialogue. Therefore, our goal is to mimic the cognitive forget function (Bodner & Lindsay, 2003) during a dialogue (i.e., reproduce the same information filtering) and to define the optimal dialogue history size to remember. We note that the inconsistency annotation is not turn-independent. For example, in order to detect that the system says the same sentence twice and the user is bothered, we need to know turns t and $t+1$.[1] A legitimate question then is: To what extent can a reduction of the dialogue history size, possibly combined with the knowledge of the user's next turn, allow the system to better detect its own inconsistent behavior?

4 Data and Experiments

4.1 The bAbI Corpus and Our Additional Annotation

To answer the question asked in Section 3, we use the bAbI dialogue corpus (Bordes *et al.*, 2017), which is a postprocessed version of the DSTC2 corpus (Henderson *et al.*, 2014a), consisting of 3,232 English dialogues between a human and a POMDP-based restaurant reservation system (Young *et al.*, 2013). Dummy API calls were added to simulate access to an external database. A dialogue turn contains either an exchange between the user and the system, or an API call and its result. Detailed statistics are provided in Henderson *et al.* (2014b). Based on the inconsistency types identified in Section 3, we automatically added inconsistency annotation to each dialogue turn by employing simple pattern-matching rules.[2] We conduct annotation evaluation on a sample of 150 dialogue turns by two linguists (with inter-annotator agreement in terms of Cohen's kappa of 0.76). We consider the annotated dataset as a silver-standard (computer annotation with human evaluation). For the evaluation, we choose labelling accuracy as the metric to reflect the annotation performance and obtain a 0.79 accuracy score. The accuracy metric is sufficient because the number of dialogues with and without inconsistencies is not overly imbalanced. Coming from the original DSTC2 corpus, each dialogue is also annotated according to the DSTC2 handbook guidelines[3] with a success mark on a satisfaction scale from 0 (unsatisfied) to 5 (satisfied) (Walker *et al.*, 1997). In total, 502 dialogues are failed (16%).

dialogue count	success	failure
with inconsistencies	1,715	420
without inconsistencies	1,020	82

Table 2 – Number of successful and failed dialogues with and without inconsistencies in bAbI data.

1. We assume that the system initiates the dialogue. Therefore, we take the next user utterance from the same turn. This is the case for DSTC2 (see Section 4).
2. The full code for the rules is available at https://github.com/DiaSER21/consistency.
3. https://github.com/matthen/dstc/blob/master/handbook.pdf

4.2 Correlation Between Failure and Inconsistencies

Unsurprisingly, almost all failed dialogues contain inconsistent system responses. The Fisher exact test (Fisher, 1936) shows that there is a very likely dependence between failed dialogues and the presence of inconsistency – dialogues with inconsistencies are ca. 3x more likely to fail (odds ratio 0.328, $p <$ 1e-20). [4] Most failed dialogue contain inconsistencies, but a much lower proportion of successful dialogues has them. Moreover, the number of inconsistencies in a dialogue is higher on average for the failed dialogues. There are many dialogues (around 15%) which can be considered as failed on closer inspection even though they are marked as successful. [5] This can explain why so many dialogues annotated in the original data as successful contain inconsistencies. The dialogue success is impacted not just by the presence of an inconsistency, but also by its relative position with respect to the key events in the transaction (e.g., API system call for fetching an answer, query for a confirmation etc.). This is why we felt justified in trying to gauge this impact.

We investigated which were the determining features in deciding whether a dialogue was a failure or not. We used Gaussian naïve Bayes (Chen *et al.*, 2009) from Scikit-Learn (Pedregosa *et al.*, 2011) to predict dialogue success. [6] Table 3 summarizes some of the different features used to improve the detection of failed dialogues. [7] If the dialogue contains inconsistencies already, they are more likely to occur again. We noticed that the types of inconsistencies are not that important to detect failed dialogues. We calculate unsuccessful dialogues' detection F1-score (unsuccessful counts as positive). The best results are achieved with simple TF-IDF-based textual features of user, system and API call inputs, coupled with the number of total inconsistencies and with the number of inconsistencies appearing before the first system's API call in the dialogue. The results confirm that inconsistencies have an influence on dialogue success.

features	precision	recall	F1-score
textual	0.56	0.52	0.53
textual + total inconsistencies	0.57	0.62	0.60
textual + total inconsistencies + inconsistencies before API call	**0.65**	0.57	**0.61**

Table 3 – Failed dialogue prediction with and without inconsistency annotation.

4.3 Models, Metrics and Experiments

Our rule-based automatic annotation (see Section 4.1) uses the whole annotated dialogue. However, we are not able to see future context in real use cases. Therefore, we raise a question about the possibility to match the performance by training a classification model based solely on the past context. We trained four different classifiers on our annotation to predict inconsistencies:

4. We note that although the dialogue inconsistencies are correlated with a higher chance of a dialogue failure, the correlation does not imply a strict cause-effect relationship, as users may be sufficiently motivated to put up with punctual inconsistencies if they feel that they can obtain what they want from the system.

5. For instance, the user never speaks during the dialogue, user requests are not satisfied, the system was unable to finish the dialogue, or there are numerous speech recognition errors.

6. Gaussian naïve Bayes worked better than other machine learning algorithms such as SVM, logistic regression, random forest and multilayer perceptron in our preliminary experiments.

7. Features used: the user and system utterances transformed into word-based TF-IDF weights, system database API call with the same TF-IDF, total number of inconsistencies in the dialogue, number of inconsistencies happening before and after the API call, types of inconsistencies present.

	Bi-LSTM				DIET				WMM2Seq				GPT-2			
	binary		multi		binary		multi		binary		multi		binary		multi	
input	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1
c	59.0	52.3	83.9	9.53	85.7	66.6	83.4	52.6	86.0	50.1	83.7	35.7	64.2	77.2	62.1	8.1
c+h_1	80.5	65.0	84.0	10.2	85.2	58.4	83.1	43.0	86.9	48.1	82.6	41.8	84.3	90.5	84.2	38.8
c+h_2	77.0	59.8	84.8	9.18	83.2	55.2	82.6	50.0	**87.0**	46.2	82.6	39.6	85.2	91.0	85.1	40.7
c+h_f	71.0	48.9	85.4	6.57	79.3	53.7	83.6	50.7	85.9	44.4	82.6	44.6	99.9	99.9	**98.4**	**93.7**
c+n	79.2	71.1	80.2	9.57	89.7	64.1	**88.9**	54.1	**90.2**	57.3	85.1	26.1	25.2	8.2	10.2	22.9
c+n+h_1	88.0	82.1	84.2	8.48	87.9	76.6	**87.6**	52.0	**89.1**	53.4	81.3	39.3	80.6	87.7	79.2	40.8
c+n+h_2	87.4	81.4	81.8	8.41	89.0	77.6	85.6	40.2	**89.1**	55.1	81.4	39.6	85.2	90.9	85.1	40.6
c+n+h_f	79.0	65.9	85.7	5.72	87.0	70.2	86.2	48.6	86.4	46.9	82.6	40.8	**99.9**	99.9	**98.5**	**93.5**

Table 4 – Inconsistency classification accuracy and weighted-averaged F1 scores (binary and multiclass mode) of our models. The most frequent baseline achieves accuracy 87%. We present results with various combinations of the input data. Possible inputs are: current turn (c), next user utterance (n), last 1 or 2 turns of dialogue history (h_1,h_2) or full history (h_f).

— **Bi-LSTM with attention** (Jang *et al.*, 2020) is a simple model for sequence/text classification but highly effective when it has to deal with long-term information such as dialogue history.
— **DIET classifier**, the dialogue intents entities transformer, is a transformer-based (Devlin *et al.*, 2019) dialogue intents classifier (Wu *et al.*, 2020) that outperforms most of recent classifiers in the user intention detection task.
— **WMM2Seq** (Chen *et al.*, 2019) is a memory network-based model that uses two different memory modules: context (dialogue history as episodic memory) and knowledge base (API calls as semantic memory) for generating system responses, one word at a time.
— **GPT-2** (Radford *et al.*, 2019) is a transformer-based architecture made of several transformer decoder blocks (Vaswani *et al.*, 2017), stacked one on top of the other. The architecture is pre-trained for language modeling on a huge corpus and is capable of effective finetuning for many downstream tasks. We finetune the model in a multitask setting, i.e. we optimize both inconsistency classification loss and response generation loss.

We use classification accuracy and weighted macro average of F1 scores as the evaluation metrics, and we train the models both for binary (inconsistency or not) and multiclass classification (predicting specific inconsistency type, or no inconsistency). We use the most frequent label prediction as a strong baseline (no inconsistency, present in 87% of the examples, i.e. accuracy 87%). The results are shown in Table 4 and discussed next. We use 2,117 dialogues for training and 1,115 for testing.

The results show that, even if the baseline is strong (87%), it is outperformed by all the models. The best results (99%) are obtained by the GPT-2 based model when using the whole dialogue history (*h*) and the next user utterance (*nu*). When *h* is not used, the performance decreases; *nu* has a smaller effect. We believe that GPT-2 is capable of extracting input information that is most relevant for inconsistency classification, therefore it benefits from the long history. Indeed, when we examined the results, we observed that almost all GPT classification errors are related to the "incompleteness" inconsistency. These cases depend only on the immediate context (previous utterance). On the contrary, DIET and WMM2Seq obtained the best results (0.90) with the next user utterance and no history at all, even if the performance difference without the next user utterance is smaller than GPT-2's. Also, a simple Bi-LSTM outperforms the baseline when using both *h1* and *h2* with *nu* in binary mode but fails to pick up the necessary features in the multiclass mode.We observe that with

model	**bi-LSTM**	**DIET**	**WMM2Seq**	**GPT-2**
κ	0.74	0.76	0.67	**0.97**

Table 5 – Cohen's Kappa values for comparing the best models' predictions to the ground truth labels.

less information, WMM2Seq gets the highest accuracy after GPT-2. However, as GPT-2 training is both costly and needs the whole dialogue to get the best performance, the results confirm the need of predicting next user's utterance to have a more accurate model, in case where a smaller model is required or when the whole dialogue history is not available. We also compare the best model variants' predictions to ground-truth labels in binary mode and measure Cohen's Kappa (Ben-David, 2008) to asses that the models' performance is better than chance. The results are shown in Table 5.

5 Conclusion and perspectives

This work presents a new dataset based on the DSTC2/bAbI corpus that allows research on the task of detecting dialogue inconsistency, which has not been explored much so far. We conducted experiments that revealed a correlation between system turn inconsistencies and dialogue failures. This fact can be exploited in further research of dialogue modeling to prevent failures. Furthermore, we applied four different classifier architectures to automatically detect inconsistencies in the newly formed dataset. Among the explored architectures, the best performing were a GPT-2-based classifier and the WMM2Seq model. Interestingly, while GPT-2 strongly benefits from the provided history context, the WMM2Seq performed best when no history was used and next user utterance was available to the model, which makes it more suitable for the real world usecases. Access to the next utterance improved results across the board. With this set of experiments, we provide a first proof of the benefit we might gain by having dialogue systems to incorporate an oracle for predicting the next user turn, a step toward a future dialogue architecture with a dual system and user model. In future works, we will evaluate on more complex datasets in order to confirm the usefulness of this new feature when detecting system inconsistencies.

Acknowledgements

This work is supported by the HumanE-AI-Net project (EC Horizon 2020 grant no. 952026) [8] and AKIO Software [9] in the form of the DIASER microproject (*WP3 – Human AI Collaboration and Interaction*) and by Charles University grants PRIMUS 19/SCI/10, GAUK 302120, and SVV 260 575. We also want to thank the reviewers for their great remarks that helped us improving greatly the paper.

References

AUGUSTE J., BÉCHET F., DAMNATI G. & CHARLET D. (2019). Skip Act Vectors: integrating dialogue context into sentence embeddings. In *Tenth International Workshop on Spoken Dialogue*

8. https://www.humane-ai.eu
9. https://www.akio.com

Systems Technology, Syracuse, Italy. HAL : hal-02125259.

BEN-DAVID A. (2008). Comparison of classification accuracy using cohen's weighted kappa. *Expert Systems with Applications*, **34**(2), 825–832.

BODNER G. E. & LINDSAY D. S. (2003). Remembering and knowing in context. *Journal of Memory and Language*, **48**(3), 563–580.

BORDES A., BOUREAU Y.-L. & WESTON J. (2017). Learning end-to-end goal-oriented dialog.

CHEN J., HUANG H., TIAN S. & QU Y. (2009). Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, **36**(3), 5432–5435.

CHEN X., XU J. & XU B. (2019). A Working Memory Model for Task-oriented Dialog Response Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 2687–2693, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1258.

CHERNYSHOVA E. (2018). *Expliciter et inférer dans la conversation : modélisation de la séquence d'explicitation dans l'interaction*. Theses, Université de Lyon. HAL : tel-02070720.

DEPPERMANN A. (2018). Inferential practices in social interaction: A conversation-analytic account. *Open Linguistics*, **4**(1), 35 – 55. DOI : https://doi.org/10.1515/opli-2018-0003.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota: Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DZIRI N., KAMALLOO E., MATHEWSON K. & ZAIANE O. (2019). Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 3806–3812, Minneapolis, Minnesota: Association for Computational Linguistics. DOI : 10.18653/v1/N19-1381.

ENGELBRECHT K.-P. & MÖLLER S. (2010). Sequential classifiers for the prediction of user judgments about spoken dialog systems. *Speech Communication*, **52**(10), 816 – 833. DOI : https://doi.org/10.1016/j.specom.2010.06.004.

FERNANDEZ R., LUCHT T., RODRIGUEZ K. & SCHLANGEN D. (2006). Interaction in task-oriented human-human dialogue: the effects of different turn-taking policies. In *2006 IEEE Spoken Language Technology Workshop*, p. 206–209. DOI : 10.1109/SLT.2006.326791.

FISHER R. A. (1936). Design of experiments. *British Medical Journal*, **1**(3923), 554–554. PMC2458144[pmcid].

GAO S., SETHI A., AGARWAL S., CHUNG T. & HAKKANI-TUR D. (2019). Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, p. 264–273, Stockholm, Sweden: Association for Computational Linguistics. DOI : 10.18653/v1/W19-5932.

HAM D., LEE J.-G., JANG Y. & KIM K.-E. (2020). End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 583–592, Online: Association for Computational Linguistics.

HANCOCK B., BORDES A., MAZARE P.-E. & WESTON J. (2019). Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association*

for Computational Linguistics, p. 3667–3684, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1358.

HENDERSON M., THOMSON B. & WILLIAMS J. D. (2014a). The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, p. 263–272, Philadelphia, PA, U.S.A.: Association for Computational Linguistics. DOI : 10.3115/v1/W14-4337.

HENDERSON M., THOMSON B. & WILLIAMS J. D. (2014b). The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, p. 263–272, Philadelphia, PA, U.S.A.: Association for Computational Linguistics. DOI : 10.3115/v1/W14-4337.

HORI C., PEREZ J., HIGASHINAKA R., HORI T., BOUREAU Y.-L., INABA M., TSUNOMORI Y., TAKAHASHI T., YOSHINO K. & KIM S. (2019). Overview of the sixth dialog system technology challenge: DSTC6. *Computer Speech & Language*, **55**, 1–25. DOI : https://doi.org/10.1016/j.csl.2018.09.004.

HOSSEINI-ASL E., MCCANN B., WU C.-S., YAVUZ S. & SOCHER R. (2020). A Simple Language Model for Task-Oriented Dialogue. *arXiv:2005.00796 [cs]*.

JANG B., KIM M., HARERIMANA G., KANG S.-U. & KIM J. W. (2020). Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. *Applied Sciences*, **10**(17). DOI : 10.3390/app10175841.

KUMAR V. (2010). Customer relationship management. *Wiley international encyclopedia of marketing*.

LAI T. M., HUNG TRAN Q., BUI T. & KIHARA D. (2020). A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 8034–8038. DOI : 10.1109/ICASSP40776.2020.9053975.

LI M., ROLLER S., KULIKOV I., WELLECK S., BOUREAU Y.-L., CHO K. & WESTON J. (2020). Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4715–4728, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.428.

LIN Z., KANG X., LI G., JI F., CHEN H. & ZHANG Y. (2020). "wait, i'm still talking!" predicting the dialogue interaction behavior using imagine-then-arbitrate model.

LITMAN D., SWERTS M. & HIRSCHBERG J. (2006). Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics*, **32**(3), 417–438. DOI : 10.1162/coli.2006.32.3.417.

MA M., HUANG L., XIONG H., ZHENG R., LIU K., ZHENG B., ZHANG C., HE Z., LIU H., LI X., WU H. & WANG H. (2019). Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3025–3036, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1289.

MADOTTO A., WU C.-S. & FUNG P. (2018). Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1468–1478, Melbourne, Australia: Association for Computational Linguistics. DOI : 10.18653/v1/P18-1136.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V. *et al.* (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, **12**, 2825–2830.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

SAMMUT C. & WEBB G. I., Éds. (2010). *POMDPs*, In C. SAMMUT & G. I. WEBB, Éds., *Encyclopedia of Machine Learning*, p. 776–776. Springer US: Boston, MA. DOI : 10.1007/978-0-387-30164-8_642.

SHANG G., TIXIER A. J.-P., VAZIRGIANNIS M. & LORRÉ J.-P. (2020). Speaker-change aware crf for dialogue act classification.

SUKHBAATAR S., SZLAM A., WESTON J. & FERGUS R. (2015). End-to-end memory networks. In C. CORTES, N. D. LAWRENCE, D. D. LEE, M. SUGIYAMA & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 28*, p. 2440–2448. Curran Associates, Inc.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, p. 5998–6008, Long Beach, CA, USA.

VERON M. (2019). Lifelong learning et systèmes de dialogue : définition et perspectives. In *Rencontres des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, Toulouse, France. HAL : hal-02301064.

WALKER M. A., LITMAN D. J., KAMM C. A. & ABELLA A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, p. 271–280, Madrid, Spain: Association for Computational Linguistics. DOI : 10.3115/976909.979652.

WELLECK S., WESTON J., SZLAM A. & CHO K. (2019). Dialogue Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3731–3741, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1363.

WESTON J., CHOPRA S. & BORDES A. (2015). Memory networks. In *3rd International Conference on Learning Representations (ICLR2015)*, San Diego, CA, USA.

WHITNEY D., ROSEN E., MACGLASHAN J., WONG L. L. S. & TELLEX S. (2017). Reducing errors in object-fetching interactions through social feedback. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, p. 1006–1013. DOI : 10.1109/ICRA.2017.7989121.

WOLTERS M., GEORGILA K., MOORE J. D., LOGIE R. H., MACPHERSON S. E. & WATSON M. (2009). Reducing working memory load in spoken dialogue systems. *Interact. Comput.*, **21**(4), 276–287. DOI : 10.1016/j.intcom.2009.05.009.

WU C.-S., HOI S. C., SOCHER R. & XIONG C. (2020). TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 917–929, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.66.

YOUNG S., GAŠIĆ M., THOMSON B. & WILLIAMS J. D. (2013). POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, **101**(5), 1160–1179. DOI : 10.1109/JPROC.2012.2225812.

ZHANG W., FENG Y., MENG F., YOU D. & LIU Q. (2019). Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4334–4343, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1426.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Évaluation de méthodes et d'outils pour la lemmatisation automatique du français médiéval

Cristina G. Holgado[1] Alexei Lavrentev[2] Mathieu Constant[3]
(1) Université de Strasbourg, F-67081, Strasbourg
(2) IHRIM, CNRS, ENS de Lyon, F-69007 Lyon, France
(3) ATILF, Université de Lorraine, CNRS, F-54063, Nancy
cristina.gholgado@gmail.com, alexei-lavrentev@ens-lyon.fr,
mathieu.constant@univ-lorraine.fr

RÉSUMÉ

Pour les langues historiques non stabilisées comme le français médiéval, la lemmatisation automatique présente toujours des défis, car cette langue connaît une forte variation graphique. Dans cet article, nous dressons un état des lieux de la lemmatisation automatique pour cette langue en comparant les performances de quatre lemmatiseurs existants sur un même jeu de données. L'objectif est d'évaluer où se situent les nouvelles techniques de l'apprentissage automatique par rapport aux techniques plus traditionnelles s'appuyant sur des systèmes de règles et lexiques, en particulier pour la prédiction des mots inconnus.

ABSTRACT

Evaluation of methods and tools for automatic lemmatization in Old French.

For non-stabilized historical languages such as old French, automatic lemmatization still presents challenges, as this language has a strong graphic variation. In this article we benchmark automatic lemmatization for this language by comparing the performances of four existing lemmatizers on the same dataset. Our goal is to evaluate where the new techniques of machine learning stand in regards to more traditional rule- and lexicon-based ones, especially for unknown words.

MOTS-CLÉS : lemmatisation, étiquetage morphosyntaxique, linguistique historique, français médiéval.

KEYWORDS: lemmatization, part-of-speech tagging, historic linguistics, Old French.

1 Introduction

La lemmatisation automatique a été l'objet de très nombreuses recherches dans le domaine du traitement automatique des langues (TAL). Elle consiste à produire, pour chaque occurrence de mots, leur forme de base telle qu'on peut la trouver dans des dictionnaires. Les nombreux outils qui sont désormais disponibles ont permis de populariser cette tâche pour un nombre important de langues.

Dans cet article, nous nous intéressons à la lemmatisation automatique du français médiéval qui fait face à de nombreux défis. L'absence de normalisation graphique rend la lemmatisation pour une forme plus difficile car elle peut apparaître sous plusieurs graphies. Le français médiéval se caractérise par une complexité morphologique plus importante que le français moderne et il est particulièrement

Volume 1 : conférence principale, pages 153–161.

marqué par la variation dialectale. Plusieurs corpus textuels lemmatisés, dictionnaires électroniques et lexiques morphologiques associés existent pour cet état de langue (Lavrentiev *et al.*, 2017). Cependant différents dictionnaires de référence utilisent parfois différentes formes d'entrée pour le même lexème, puisque certains privilégient les formes modernisées et d'autres conservent les formes médiévales. Il est par conséquent nécessaire de normaliser les lemmes avant de compiler un corpus d'apprentissage unifié.

Nous souhaitons dresser un état des lieux de la lemmatisation automatique du français médiéval en comparant différents outils sur un même ensemble de textes. Il existe deux grandes familles d'approches pour la lemmatisation : (1) les approches traditionnelles se fondant sur des lexiques et des règles (Silberztein, 1994; Beesley & Karttunen, 2003), (2) les approches reposant sur des techniques d'apprentissage automatique à partir de corpus annotés (Chrupala *et al.*, 2008; Müller *et al.*, 2015; Bergmanis & Goldwater, 2018), parfois complémentés de lexiques. Nous cherchons ici à évaluer où se situent les nouvelles techniques de l'apprentissage automatique par rapport aux techniques plus traditionnelles et éprouvées dans le cadre de la lemmatisation du français médiéval. Plus précisément, nous allons tester quatre outils : (1) TreeTagger (Schmid, 1994) ; (2) LGeRM (Souvay & Pierrel, 2009) ; (3) Pie (Manjavacas *et al.*, 2019) ; (4) UDPipe (Straka & Straková, 2017). Les trois premiers sont très utilisés pour la lemmatisation du français médiéval. Ils utilisent des approches assez différentes donc ils sont intéressants à comparer. Le dernier est un outil très populaire dans la communauté TAL actuelle, utilisant une méthode de lemmatisation encore différente des trois premiers.

L'article est organisé comme suit. Tout d'abord, nous présentons les outils utilisés. Ensuite, nous précisons les caractéristiques du corpus et la procédure expérimentale. Enfin, nous proposons une discussion des résultats et des pistes pour améliorer la lemmatisation du français médiéval.

2 Description des lemmatiseurs utilisés

Nous décrivons maintenant les quatre outils de lemmatisation testés dans l'article pour le français médiéval. Les outils TreeTagger et LGeRM sont des systèmes à base de règles et de lexiques pour la lemmatisation. TreeTagger est, avant tout, un étiqueteur morphosyntaxique basé sur des arbres de décision appris à partir d'un corpus annoté. Il existe un module de lemmatisation qui consiste à récupérer le (ou les) lemme(s) du mot en entrée dans le lexique du corpus d'apprentissage et potentiellement d'un lexique externe. La prédiction de l'étiquette grammaticale permet de filtrer les lemmes du lexique qui ne sont pas de cette catégorie. Pour les mots inconnus du lexique, si l'option correspondante est activée, le lemme prédit correspond simplement à la copie du mot en entrée. La version récente de LGeRM repose sur un principe similaire à quelques différences notables près. Tout d'abord, il utilise TreeTagger pour prédire l'étiquette morphosyntaxique du mot à lemmatiser. Ensuite, il repose sur le lexique extrait (principalement) du Dictionnaire du Moyen Français (DMF) (Bazin-Tacchella *et al.*, 2016). Enfin, un système de règles complexes est appliqué pour prédire le (ou les) lemme(s) des mots inconnus du lexique. Les outils Pie et UDPipe s'appuient sur un apprentissage supervisé de leurs modèles de lemmatisation à partir de corpus annotés. Pie utilise un modèle neuronal encodeur-décodeur appris conjointement avec une tâche de prédiction des mots suivants et précédents afin de mieux tenir compte du contexte, sans avoir à prédire l'étiquette morphosyntaxique. L'outil, incluant également un étiqueteur morphosyntaxique, a démontré des résultats très prometteurs pour la lemmatisation des états anciens de différentes langues. UDPipe, quant à lui, incorpore un autre

type de lemmatiseur appris sur corpus annoté. Étant donné un mot à lemmatiser, le principe est de générer un ensemble de paires (lemme, étiquette morphosyntaxique) possibles à l'aide de règles de lemmatisation apprises automatiquement, puis d'appliquer un modèle de levée d'ambiguïté appris grâce à un perceptron moyenné.

3 Campagne de tests

3.1 Source des données et normalisation

Les textes annotés utilisés à l'entraînement et à l'évaluation sont issus du corpus BFMGOLDLEM rassemblant un total de 431 144 formes étiquetées et lemmatisées. Ce corpus fait partie de la BFM (Base de Français Médiéval) [1] (Guillot *et al.*, 2018), une collection de textes médiévaux qui recouvre la période du IX^{e} jusqu'au XV^{e} siècle et dont le nombre total d'occurrences-mots s'élève à 4,7 millions. Il est composé de deux sources dont une prédominante appartient à un seul auteur (Chrétien de Troyes). Il a été lemmatisé à l'ATILF dans le cadre du projet DECT (Souvay & Kunstmann, 2008). C'est donc un corpus important, mais peu diversifié (un seul auteur, un seul manuscrit, un seul genre). Il a son propre référentiel de lemmes, qui correspondent pour la plupart aux entrées du dictionnaire Tobler-Lommatzsch (TL) (Adolf, 2002) qui privilégie des formes anciennes. Le reste du corpus a été lemmatisé dans le cadre de la BFM et est beaucoup plus diversifié. Il est au format CONLL-U (Nivre *et al.*, 2016), et constitué par les formes fléchies tokenisées, accompagnées des étiquettes morphologiques du jeu d'étiquettes Cattex09 (Prévost *et al.*, 2009) et des lemmes. Ces lemmes sont majoritairement issus du DMF, qui utilise des formes modernes. D'autres référentiels ont été utilisés en cas d'absence de lemme dans le DMF.

Afin d'harmoniser dans la mesure du possible les lemmes dans le corpus d'apprentissage, les lemmes DECT dans les textes de Chrétien de Troyes, ont été convertis en lemmes d'autres référentiels à partir du lexique FROLEX (Lavrentiev *et al.*, 2017) selon le procédé suivant : conversion vers un lemme DMF (s'il existe), sinon TL, sinon GDF (Godefroy, 1901). Le lemme DECT a été conservé en cas d'absence de correspondance. Dans le corpus normalisé, les 424 836 lemmes sont ainsi des lemmes DMF (98,54%), 4 512 DECT (1,04%), 965 BFM (0,22%) [2], 801 TL (0,18%) et 30 GDF (0,01%).

3.2 Protocole de tests

Afin d'évaluer les outils, nous avons mis en place un protocole de tests se voulant le plus réaliste possible. Il est divisé en dix expériences d'évaluation incluant des corpus de textes de caractéristiques linguistiques et tailles différentes, dont certains sont plus marqués dialectalement et/ou datant d'états plus anciens de la langue, ce qui arrive effectivement dans la pratique. Ainsi, notre corpus a été divisé en dix parties, chacune contenant un ou plusieurs textes cohérents en termes de date, de genre et de dialecte. Pour chaque expérience, les outils (étiquetage et/ou lemmatisation) étaient appris sur tout le corpus, sauf la partie testée (que l'on appelle corpus de contrôle). Le découpage est indiqué dans le tableau 2. Le tableau 1 indique les caractéristiques des textes de contrôle pour chaque test. Une

1. http://txm.bfm-corpus.org
2. Les lemmes BFM sont essentiellement des noms propres du corpus BFM. Aucun lemme DECT n'a été converti vers ce référentiel.

description plus détaillée du protocole de tests est accessible dans un dépôt GIT du projet [3].

Test	Date	Dialecte	Genre
1 et 2	fin 12e s.	champenois	roman
3	milieu 11e s.	normand	hagiographie
4	début 12e s.	normand	épique
5	milieu 12e s.	anglo-normand	chronique
6	début 14e s.	non marqué	chronique
7	fin 13e s.	non marqué	hagiographie
8	début 11e s.	franco-occitan	hagiographie
9	milieu 13e s.	hainaut	charte
10	fin 14e s.	non marqué	registre

TABLE 1: Métadonnées caractéristiques des textes de contrôle pour chaque test

La première partie (test 1) contient tous les textes de Chrétien de Troyes, ce qui constitue le corpus d'apprentissage avec la taille la plus réduite (41,1%) par rapport aux autres tests, pour lesquels le corpus de contrôle constitue entre 1,2% et 8,2% du corpus total en nombre de tokens. Cela permet d'évaluer l'impact du retrait du corpus d'apprentissage d'un volume de textes important mais très homogène. Dans le test 2, un seul texte de Chrétien de Troyes constitue le corpus de contrôle, les autres textes de cet auteur font partie du corpus d'apprentissage. Il convient de noter que la proportion des formes inconnues varie très fortement selon les tests et ce n'est pas dans le premier test qu'elle est la plus élevée (11,40% contre 31,85% dans le test 8 où le corpus de contrôle est composé de deux textes courts, mais très anciens et marqués dialectalement). Dans les tests 3 et 4, les textes de contrôle sont également très anciens, mais les graphies sont moins inhabituelles. Dans les tests 9 et 10, les textes de contrôle ne sont pas marqués sur le plan diachronique ou dialectal, mais appartiennent à un genre peu représenté dans le corpus d'apprentissage (acte juridique).

3.3 Configuration des outils

TreeTagger a été utilisé sans lexique externe. Cela implique que le lexique utilisé est celui du corpus d'apprentissage. Nous utilisons la dernière version de LGeRM qui nous a été fourni par le développeur de l'outil. Cette version repose sur l'étiqueteur TreeTagger pré-appris. Pour UDPipe, nous avons utilisé la bibliothèque associée dans R (UDPipe v0.8.3). L'entraînement du lemmatiseur est appris en utilisant 60 itérations (epochs), un batch de taille 100 et un taux d'apprentissage (learning rate) de 0,1. L'étiqueteur est appris en utilisant 20 itérations. Les autres paramètres de configuration correspondent aux valeurs par défaut. Ceux de Pie (v0.8.5) ont été fixés de la manière suivante : 50 itérations (epochs) et un batch de taille 50 (lemmatiseur) et 10 itérations et un batch de taille 50 (étiqueteur). Le lemmatiseur et l'étiqueteur utilisent tous deux un taux d'apprentissage de 0,0001.

4 Analyse des résultats

Le tableau 2 affiche la précision moyenne (micro) obtenue pour chaque outil sur l'ensemble des tests.

3. https://gitlab.huma-num.fr/lemmatisation-fro/bfm-lem

La ponctuation n'est pas prise en compte dans le calcul. Il comprend, d'une part, la précision obtenue sur l'ensemble des lemmes et, d'autre part, celle obtenue pour les mots inconnus, c'est-à-dire, les unités absentes du corpus d'apprentissage (CA). Du fait que LGeRM peut proposer plusieurs lemmes, il a été évalué différemment en divisant le nombre de lemmes corrects par le nombre de lemmes proposés pour la forme. Ses résultats sur les mots inconnus doivent être regardés avec précaution car l'outil ne s'appuie pas sur le corpus d'apprentissage mais sur un lexique externe. Ainsi, l'évaluation des ces mots inclut des mots présents dans son lexique.

T	**CA**		**CC**			**TreeTagger**		**LGeRM**		**UDPipe**		**Pie**	
	tokens	%	tokens	m.inc.	%	tout	inc.	tout	inc.	tout	inc.	tout	inc.
1	177 050	41,1	254 094	28 976	11,4	0,75	0,07	0,83	0,82	0,67	0,12	0,74	**0,36**
2	383 164	88,9	47 965	1275	2,6	**0,86**	0,09	0,83	**0,84**	**0,76**	0,19	0,71	0,22
3	425614	98,7	5530	770	13,9	0,73	0,07	0,78	0,56	0,65	0,09	0,60	0,21
4	395 832	91,8	35 312	5399	15,3	0,72	0,07	0,85	0,75	0,70	0,13	0,61	0,20
5	413 123	95,8	18 021	2313	18,8	0,77	0,10	0,86	0,63	0,69	0,12	0,69	0,26
6	420 109	97,4	11 035	1405	12,7	0,81	0,20	**0,90**	0,71	0,71	0,20	0,69	0,23
7	408 375	94,7	22 769	2008	8,81	0,79	0,12	0,89	0,77	0,73	0,20	**0,75**	0,31
8	426 052	98,8	5092	1622	31,8	0,43	0,02	0,61	0,39	0,42	0,06	0,45	0,16
9	420 652	97,6	10 492	1711	16,3	0,74	0,13	0,82	0,54	0,68	0,09	0,67	0,16
10	419 163	97,2	11 981	2380	19,9	0,76	**0,29**	**0,90**	0,77	0,64	**0,21**	0,66	0,17
	Moyenne					0,74	0,12	0,83	0,68	0,66	0,14	0,66	0,23

TABLE 2: Découpage des expériences (CA corpus d'apprentissage ; CC corpus de contrôle). Moyenne des tests (hors ponctuation) (précision)

Nous observons que les meilleurs résultats coïncident pour les quatre outils sur un groupe spécifique de tests, en particulier les tests 2, 6 et 7. Les tests 2 et 7 correspondent effectivement aux textes contenant le plus petit pourcentage de mots inconnus. En ce qui concerne l'ensemble des tokens (tout), LGeRM obtient le meilleur résultat parmi l'ensemble des outils avec une précision moyenne de 83%. Ceci s'explique en partie du fait qu'il possède un lexique très riche. Néanmoins, il est principalement conçu pour le Moyen Français ce qui explique la faible précision pour les tests 3 et 8 constitués de textes très anciens en français médiéval, et la précision la plus élevée pour les tests 6 et 10, constitués de textes plus tardifs. D'autre part, les autres outils ont été entrainés avec le même jeu de données en l'absence de lexique externe. De ce groupe, TreeTagger a atteint la précision la plus élevée avec 74% pour les lemmes, suivi par Pie avec 66% et UDPipe avec 64%. Dans le cas de TreeTagger, la taille du CA a un impact sur sa performance, ce qui n'est pas le cas dans UDPipe et Pie, comme cela est illustré dans les tests 1, 2 et 7, si nous tenons compte du pourcentage relativement faible de mots inconnus. En effet, le test 4 comporte un CA plus réduit que le test 7 mais le nombre de mots inconnus dans le CC est plus élevé. De ce fait, nous constatons que le nombre de mots inconnus a, en définitive, un impact sur l'ensemble des outils. Cependant, dans les outils UDPipe et Pie, et au contraire de TreeTagger, la taille des données n'est pas forcément significative pour l'amélioration de la précision mais plutôt le nombre assez représentatif d'échantillons dans les catégories pour obtenir une meilleure modélisation, permettant à ces outils d'être capables de prédire un lemme lorsqu'ils rencontrent de nouvelles formes. Pie est ainsi plus performant que TreeTagger dans le test 8 qui est caractérisé par deux textes très anciens et marqués dialectalement dont les tokens ne sont pas présents dans le corpus d'apprentissage. Afin d'examiner plus en détail ces résultats et mieux comprendre les performances des lemmatiseurs pour les mots inconnus, nous avons calculé la précision des lemmes pour chacune des étiquettes du corpus (cf. tableau 3).

Cat.	Tokens	%	m.inc.	%	TreeTagger		LGeRM		UDPipe		Pie	
					tout	inc.	tout	inc.	tout	inc.	tout	inc.
ADJ	14 773	4,03	2680	5,60	0,73	0,10	0,83	0,67	0,65	0,11	0,55	0,18
ADV	39 535	10,78	2435	5,09	0,71	0,10	0,63	0,75	0,81	0,13	0,62	0,18
CON	37 233	10,15	44	0,09	0,82	0,00	0,94	0,37	0,94	0,02	0,77	0,44
DET	35 853	9,77	812	1,70	0,68	0,06	0,81	0,55	0,72	0,05	0,65	0,15
Ncom	53 989	14,72	12 649	26,43	0,68	0,12	0,71	0,68	0,51	0,14	0,50	0,20
Npro	9268	2,53	6058	12,66	0,54	0,40	0,43	0,47	0,34	0,30	0,26	0,03
PRE	34 309	9,35	667	1,39	0,66	0,08	0,80	0,60	0,77	0,18	0,59	0,12
PRO	60 870	16,59	770	1,61	0,72	0,01	0,75	0,60	0,67	0,06	0,57	0,15
VER	80 522	21,95	21 413	44,74	0,62	0,02	0,84	0,80	0,57	0,13	0,60	0,36
Total	366 882		47 859									

TABLE 3: Précision des lemmes par catégorie

Tout d'abord, nous constatons que les mots inconnus appartiennent aux parties du discours qui possèdent le plus grand nombre de formes fléchies (d'abord, les verbes, puis les noms communs et les noms propres). Les performances relativement élevées de LGeRM s'expliquent par l'importance de son lexique (indépendant du corpus d'apprentissage) et le fait que les règles de substitution permettent le plus souvent de retrouver une forme attestée dans le lexique. TreeTagger prend la forme pour lemme en cas de mots inconnus lorsqu'on active l'option « -no-unknown ». Cette stratégie est assez efficace pour les noms propres, mais elle échoue systématiquement pour les verbes (à l'exception des infinitifs). Ainsi, il gagne en précision dans le cas des mots inconnus si le token et le lemme sont identiques (e.g. : le nom propre *Rome*), ce qui est rarement le cas des verbes. Ces résultats limités pour les mots inconnus peuvent également s'expliquer par le fait que, dans la très grande majorité des cas, les lemmes de référence sont modernisés, ce qui réduit encore les chances que la forme d'un mot soit identique au lemme. En particulier, l'outil obtient un score nul sur les conjonctions inconnues : ex. les formes *maiz* et *conbien* ont respectivement pour lemmes *mais* et *combien*.

Au contraire, UDPipe et Pie sont moins affectés par le nombre de formes fléchies et plus performants que TreeTagger pour quasiment toutes les catégories. Selon les résultats obtenus par ces deux outils, de manière générale, plus le nombre de tokens augmente, plus la précision augmente. Cela s'explique en partie par la capacité "générative" de ces deux outils qui sont capables de prédire une paire (forme, lemme) qui n'aura jamais été vue dans le corpus d'apprentissage (cf. section 2). La quantité d'exemples donnés à l'apprentissage est cruciale dans ce cadre-là pour que les procédures d'apprentissage puisse en extraire automatiquement des généralisations. Par exemple, dans le test 10, le pronom inconnu *luy* et le verbe inconnu *deposeroit* sont correctement lemmatisés en *lui* et *déposer* par Pie. Dans ce même test, le nom propre inconnu *Yvein* et l'adverbe inconnu *Meïsmes* sont correctement lemmatisés en *Yvain* et *même* pour UDPipe.

Pie est, en général, plus performant que UDPipe sur les mots inconnus, à l'exception toutefois des noms propres et des prépositions. Pie obtient ses performances les plus notables sur les conjonctions (44% de précision) et les verbes (36%). UDPipe est particulièrement performant sur les noms propres (27%). La différence de performances entre les deux outils est particulièrement marquante sur les conjonctions et les pronoms en faveur de Pie. Les résultats relativement plus élevés dans ces catégories s'expliquent par le fait que la plupart des tokens correspondent à un nombre très limité de types (e.g. conjonctions 'qua', 'comme', 'mais'). Certains mots grammaticaux courts et fréquents semblent avoir un impact important sur les analyses UDPipe. Par exemple, la contraction

‘du’ PRE.DETdef/de.le influence l’analyse de plusieurs formes qui se terminent en -du. Par exemple, *desfandu* : PRE.DETdef/*desfande.le*. Pie propose, dans ce cas, l’analyse correcte : VERppe/*défendre*. Une liste de catégories fermées fournie à l’outil en tant que paramètre aurait permis d’éviter ce genre d’erreur. Inversement, les noms propres obtiennent une précision particulièrement faible dans Pie relativement à UDPipe, en particulier pour les formes inconnues (3% vs. 27%).

Concernant Pie, la lemmatisation et l’étiquetage sont indépendants (cf. section 2). Cet outil produit donc parfois des analyses incohérentes du point de vue linguistique. Par exemple, pour la forme *Alexis*, l’étiquette ‘NOMpro’ est correcte, mais le lemme proposé *Aller* ressemble à un infinitif. La même situation se produit pour la forme *amfant* : NOMcom/*ampendre* (lemme correct : *enfant*). UDPipe, qui sélectionne des paires (lemme, étiquette), propose l’analyse VERppa/*amfer*, qui est erronée, mais cohérente. Nous indiquons, dans le tableau 4, quelques exemples d’erreurs de UDPipe et Pie dans la prédiction des lemmes pour des formes inconnues.

Forme	**Lemme et étiquette prédits**		**Lemme et étiquette gold**		**Outil**	**Type d’erreur**
Jehan	*Jehan*	NOMpro	*Jean*	NOMpro	Pie	Forme = lemme
Choisy	*Choisy*	NOMpro	*Choisu*	NOMpro	UDPipe	-y transformé en -u
Caisnoit	*Caisnir*	VER	*Quesnoy*	NOMpro	UDPipe	Mauvaise étiquette
dudit	*dudoulir*	VERinf	*de.ledit*	PRE.DET	Pie	Mauvaise étiquette
en.ii.	*en.ii*	DETcar	*ambedeux*	DETcar	UDPipe	Cas spécifique
Berthier	*Berthier*	NOMpro	*Berter*	VERinf	Pie	Mauvaise étiquette
Alexis	*Aller*	NOMpro	*Alexis*	NOMpro	Pie	Incohérence linguistique

TABLE 4: Quelques exemples d’erreurs sur les mots inconnus.

5 Conclusion

Dans cet article, nous avons évalué quatre outils de lemmatisation pour le français médiéval, utilisant différentes méthodes. Les résultats expérimentaux ont montré que les approches fondées sur des lexiques et systèmes de règles étaient les plus performantes du fait du manque de couverture des corpus annotés pour les méthodes supervisées. Néanmoins, les résultats sur les mots inconnus montrent que les approches par apprentissage automatique sont bénéfiques. Ces conclusions ouvrent de nombreuses pistes pour améliorer les performances. La richesse du lexique morphologique reste à ce jour le facteur déterminant dans la réussite de la lemmatisation. La première piste serait donc d’utiliser le lexique et les règles de LGeRM en association non pas avec TreeTagger, mais avec les étiqueteurs plus récents. L’augmentation du corpus d’apprentissage avec une meilleure représentation des périodes et des dialectes du français médiéval devrait également avoir un impact positif sur les performances de ces outils. L’utilisation simultanée de plusieurs outils et la mise en place d’un système de vote pour sélectionner le lemme et l’étiquette les plus probables est également très prometteuse. Le poids de chaque outil dans le vote devrait être calibré en fonction de ses performances pour chaque catégorie grammaticale. Enfin, des réglages et des post-traitements spécifiques pourraient être appliqués à chaque outil.

Remerciements

Le travail décrit dans cet article a été financé par l’Agence Nationale de la Recherche, via le projet PROFITEROLE (ANR-16-CE38-0010).

Références

ADOLF T. (2002). *Tobler-Lommatzsch : Altfranzôsisches Wôrterbuch*. Wiesbaden : Franz Steiner Verlag.

BAZIN-TACCHELLA S., MARTIN R. & SOUVAY G. (2016). *DMF 2015 - Dictionnaire du Moyen Français (version 2015)*. ATILF.

BEESLEY K. R. & KARTTUNEN L. (2003). *Finite State Morphology*. CSLI Publications. Google-Books-ID : 59RoAAAAIAAJ.

BERGMANIS T. & GOLDWATER S. (2018). Context Sensitive Neural Lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1391–1400, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1126.

CHRUPALA G., DINU G. & VAN GENABITH J. (2008). Learning Morphology with Morfette. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco : European Language Resources Association (ELRA).

GODEFROY F. (1901). *Lexique de l'ancien français*. H. Welter.

GUILLOT C., HEIDEN S. & LAVRENTIEV A. (2018). Base de français médiéval : une base de référence de sources médiévales ouverte et libre au service de la communauté scientifique. *Diachroniques. Revue de Linguistique française diachronique*, **7**, 168–184. Publisher : Presses de l'Université Paris-Sorbonne (PUPS).

LAVRENTIEV A., HEIDEN S. & DECORDE M. (2017). Building an Open Morphological Lexicon and Lemmatizing Old French Texts with the TXM Platform. In *Corpus linguistics - 2017*, Proceedings of the international conference "Corpus linguistics - 2017", p. 48–52, St-Pétersbourg, Russia : St-Petersburg State University and Institute for Linguistic Studies (RAS) and Herzen State Pedagogical University of Russia.

MANJAVACAS E., KÁDÁR A. & KESTEMONT M. (2019). Improving Lemmatization of Non-Standard Languages with Joint Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 1493–1503, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1153.

MÜLLER T., COTTERELL R., FRASER A. & SCHÜTZE H. (2015). Joint Lemmatization and Morphological Tagging with Lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 2268–2274, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1272.

NIVRE J., DE MARNEFFE M.-C., GINTER F., GOLDBERG Y., HAJIČ J., MANNING C. D., MCDONALD R., PETROV S., PYYSALO S., SILVEIRA N., TSARFATY R. & ZEMAN D. (2016). Universal Dependencies v1 : A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, p. 1659–1666, Portorož, Slovenia : European Language Resources Association (ELRA).

PRÉVOST S., GUILLOT C., LAVRENTIEV A. & HEIDEN S. (2009). Jeu d'étiquettes CATTEX 2009.

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

SILBERZTEIN M. (1994). INTEX : a corpus processing system. In *Coling'94. The 15th International Conference on Computational Linguistics. Proceedings*, volume 1, p. 579–583 : Association for Computational Linguistics. DOI : 10.3115/991886.991988.

SOUVAY G. & KUNSTMANN P. (2008). DÉCT (Dictionnaire Électronique de Chrétien de Troyes) : model for today's lexicography ? In *Proceedings of the XIII EURALEX International Congress (Barcelona, 15-19 July 2008), 2008, ISBN 978-84-96742-67-3, págs. 1203-1208*, p. 1203–1208. Section : Proceedings of the XIII EURALEX International Congress (Barcelona, 15-19 July 2008).

SOUVAY G. & PIERREL J.-M. (2009). LGeRM Lemmatisation des mots en Moyen Français. *Traitment automatique des langues*, **50**(2), 149–172.

STRAKA M. & STRAKOVÁ J. (2017). Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 88–99, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/K17-3009.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Extraction automatique de relations sémantiques d'hyperonymie et d'hyponymie dans un corpus métier

Camille Gosset[1], Mokhtar Boumedyen Billami[1], Mathieu Lafourcade[2], Christophe Bortolaso[1], Mustapha Derras[1]
(1) Berger-Levrault, Labège, France
(2) LIRMM, Université Montpellier, Montpellier, France
(1){camille.gosset, mb.billami, christophe.bortolaso, mustapha.derras}@berger-levrault.com
(2) mathieu.lafourcade@lirmm.fr

RESUME

Nous nous intéressons dans cet article à l'extraction automatique de relations sémantiques d'hyperonymie et d'hyponymie à partir d'un corpus de spécialités métier. Le corpus regroupe des ouvrages et articles en français d'expertise juridique et a été partiellement annoté en termes-clés par des experts. Nous prétraitons ces annotations afin de pouvoir les retrouver dans ce corpus et obtenir un concept général pour extraire les relations entre ces termes. Nous décrivons une étude expérimentale qui compare plusieurs méthodes de classification appliquées sur des vecteurs de relations construits à partir d'un modèle Word2Vec. Nous comparons les résultats obtenus grâce à un jeu de données construit à partir de relations d'hyperonymie tirées d'un réseau lexico-sémantique français que nous inversons pour obtenir les relations d'hyponymie. Nos résultats montrent que nous obtenons une classification pouvant atteindre un taux d'exactitude de 92 %.

ABSTRACT

Automatic extraction of hypernym and hyponym relations in a professional corpus.
In this paper, we are interested in the automatic extraction of hypernymy and hyponymy semantic relations from a corpus of business specialties. The corpus includes books and articles in French of legal expertise and has been partially annotated with keywords by experts. We pre-process these annotations to be able to find them in this corpus and to obtain a general concept to extract the relationships between these terms. We describe an experimental study which compares several classification methods applied on vectors of relations constructed using Word2Vec. We compare the results obtained using a dataset constructed from hypernymy relations drawn from a French lexico-semantic network that we invert to obtain the hyponymy relations. Our results show that we obtain a classification that can reach an accuracy rate of 92%.

MOTS-CLES : Extraction de relations d'hyperonymie et d'hyponymie, Word2Vec, réseau lexico-sémantique, apprentissage automatique, classification.

KEYWORDS: Extraction of hypernymy and hyponymy relations, Word2Vec, lexico-semantic network, machine learning, classification.

Volume 1 : conférence principale, pages 162–170.

1 Introduction

L'identification de concepts et de relations dans des documents est une phase clé dans la construction d'une base de connaissances ontologiques (Asim et al., 2018 ; Buitelaar et al., 2005). La construction manuelle d'un tel type de ressources est une tâche à forte intensité de main-d'œuvre. Bien que les données non structurées puissent être transformées en données structurées, cette construction englobe un processus très long et coûteux (Cullen & Bryman, 1988). Plusieurs travaux de recherche tendent actuellement vers l'apprentissage automatique des ontologies afin de conjurer le goulot d'étranglement d'acquisition des connaissances (Konys, 2019). L'acquisition automatique des ontologies à partir de textes est devenue ainsi un domaine de recherche prépondérant du fait de la grande masse de données présentes sur le web dont ces dernières peuvent être décrites sous-forme structurées, semi-structurées ou non structurées (Deb et al., 2018). Buitelaar et al. (2005) ont proposé une méthodologie permettant de construire des ontologies à partir de textes et cela en plusieurs étapes. L'ensemble du processus de leur méthodologie est connu sous le nom de « mille-feuille d'apprentissage de l'ontologie » (*Ontology Learning Layer Cake*). L'extraction de relations est l'une des étapes de ce processus et constitue un niveau essentiel pour la structuration des connaissances.

Plusieurs travaux de recherche se recentrent sur l'extraction de relations d'hyperonymie et d'hyponymie (par symétrie) car elles permettent d'avoir la hiérarchie de concepts dans une ontologie (Panchenko et al., 2013 ; Mintz et al., 2009 ; Bunescu & Mooney, 2005). Nous nous intéressons, dans cet article, à cette particularité d'extraction de relations de type '*Hyperonyme*' et '*Hyponyme*' au sein de textes provenant de 8 domaines du secteur public, à savoir : (1) '*État civil et Cimetières'*, (2) '*Élections'*, (3) '*Commande publique'*, (4) '*Urbanisme'*, (5) '*Comptabilité et finances locales'*, (6) '*Ressources humaines territoriales'*, (7) '*Justice'* et (8) '*Santé'*. Par définition, l'hyperonymie permet de représenter les termes pouvant être associés aux génériques d'une cible donnée, par exemple la *grippe aviaire* a pour termes génériques *grippe*, *maladie*, *pathologie* voire un *processus pathologique*. L'hyponymie, quant à elle, permet de représenter les termes pouvant être associés aux spécifiques d'une cible donnée, par exemple *engagement* a pour termes spécifiques *mariage*, *engagement à long terme*, *engagement social*, *engagement de servir* voire *engagement d'achat ferme*. Chaque relation peut faire référence à un domaine bien défini. Dans les exemples que nous venons de citer, c'est le domaine de la « santé » qui peut être identifié lorsque nous évoquons la *grippe aviaire* alors qu'il s'agit du domaine de « l'état civil » lorsque nous évoquons l'*engagement en mariage*.

Nous présentons tout d'abord dans la section 2 un état de l'art sur les méthodes d'extraction de relations. Ensuite, nous décrirons dans la section 3 les données exploitées ainsi que les prétraitements effectués sur ces données. Par la suite, dans la section 4, nous détaillons notre méthodologie pour l'extraction de relations. Enfin, nous discutons dans la section 5 de nos résultats d'expérimentation obtenus avant de conclure et présenter des perspectives de notre travail (cf. section 6).

2 Travaux antérieurs d'extraction de relations

Plusieurs travaux de recherche sur l'extraction de relations ont été proposés, deux grandes catégories d'approches existent (Granada et al., 2018 ; Wang et al., 2017), à savoir : (1) les approches à base de patrons lexico-syntaxiques (Panchenko et al., 2013 ; Hearst, 1992) ; et (2) les approches à base d'apprentissage automatique/apprentissage profond (Xue et al., 2018 ; Bunescu & Mooney, 2005).

L'approche d'extraction de relations la plus ancienne revient à Hearst (1992). Elle repose sur l'utilisation de patrons lexico-syntaxiques pour l'extraction des hyperonymes dont la langue est l'anglais. Pour la langue française, Panchenko et al. (2013) se sont intéressés aux patrons de Hearst : ils ont itéré sur l'étude et l'ont étendue afin de limiter un certain nombre de bruits. Par la suite, Panchenko et al. (2016) et Bordea et al. (2015) ont montré que ce type d'approches produit des résultats intéressants, notamment du point de vue de la précision, mais ne couvre qu'une partie de l'information. L'utilisation d'un nombre prédéfini de patrons lexico-syntaxiques pose les difficultés, d'une part, d'expliciter certaines relations et, d'autre part, de gérer l'ambiguïté de certains patrons.

D'autres approches à base d'apprentissage automatique ont été proposées afin de limiter l'utilisation des patrons lexico-syntaxiques, que ce soit par supervision (Pantel & Pennacchiotti, 2006 ; Bunescu & Mooney, 2005 ; Snow et al., 2004) ou non supervision (Mintz et al., 2009 ; Morin & Jacquemin, 2004). L'une des premières approches de cette catégorie a été proposée par Brin (1998). Elle s'appuie sur une technique de *bootstrapping* qui consiste à sélectionner des patrons par apprentissage semi-supervisé afin de construire une base d'apprentissage. Sur la même idée, des techniques de sélection de mots par analyse distributionnelle et de sélection de traits sémantiques ont été utilisées pour identifier des relations entre des entités nommées (Etzioni et al., 2004).

Par ailleurs, Cartier (2015) a proposé une approche hybride d'extraction de relations à partir d'un corpus d'un million de définitions provenant de deux ressources, à savoir : le TLFi, *Trésor de la Langue Française informatisé* (Dendien & Pierrel, 2003) et Wikipédia. Cette approche combine la précision des patrons lexico-syntaxiques et le rappel des méthodes statistiques par analyse distributionnelle. Plusieurs relations sémantiques ont été étudiées dont l'hyperonymie et l'hyponymie font partie. Le but du travail mené par Cartier (2015) est d'obtenir automatiquement une ressource sémantique pour le français contemporain à partir d'un corpus de textes.

Hashimoto et al. (2015) se sont intéressés à classifier les relations sémantiques en utilisant des plongements lexicaux (*Word Embeddings*). Dans le même principe, Mallart et al. (2020) ont proposé une approche d'identification de relations dans un corpus métier du journalisme par une modélisation LSTM (*Long Short-Term Memory*) et une utilisation de plongements lexicaux pré-entraînés avec un Word2Vec et une architecture Skip-Gram (Mikolov et al., 2013). Un modèle de classification binaire a été aussi proposé et a permis d'améliorer significativement les résultats obtenus.

Dans cet article, nous nous positionnons dans un travail d'extraction de relations par apprentissage automatique et à partir d'un corpus de données regroupant un ensemble de domaines métier. Cet apprentissage est guidé par un grand réseau lexico-sémantique où les sens communs/métiers sont définis de base. Notre objectif est un peu similaire à celui de Cartier (2015) puisque nous nous intéressons à produire des bases ontologiques métiers. De même, les travaux de Mallart et al. (2020) et Hashimoto et al. (2015) sont proches du nôtre puisque des plongements lexicaux et des modèles de classification sont utilisés. Cependant, l'originalité du travail que nous proposons dans cet article est double : (1) une extraction de relations sémantiques sans avoir besoin d'un corpus annoté sémantiquement (sens/relations) et (2) une prise en compte de tous les termes à classe ouverte (noms communs, entités nommés, adjectifs, adverbes et verbes).

3 Données de travail

Nous utilisons un corpus français contenant 172 ouvrages et 12 838 articles en ligne d'une expertise juridique et pratique. L'ensemble des documents de ce corpus est édité par la société Berger-Levrault.

Ce corpus traite 8 domaines de spécialité, à savoir : (1) '*État civil et Cimetières'*, (2) '*Élections'*, (3) '*Commande publique'*, (4) '*Urbanisme'*, (5) '*Comptabilité et finances locales'*, (6) '*Ressources humaines territoriales'*, (7) '*Justice'* et (8) '*Santé'*. Dans ce qui suit, nous faisons référence au corpus par le nom MÉTIER. Ce dernier a été partiellement annoté par des experts. Chaque paragraphe de chaque document (ouvrage ou article) est annoté avec des termes-clés. Cela constitue une représentation semi-structurée par le biais de balises HTML. Concrètement, plus de 45 000 annotations manuelles en termes-clés ont été effectuées. Par exemple, dans la phrase *« Démocratie de proximité, l'expression suscite immédiatement un doute, une inquiétude, un trouble […] »,* le terme *démocratie de proximité* est considéré comme terme-clé. Autre exemple, *« Une réforme à la recherche d'une démocratie participative à l'échelon local »*, ici *démocratie participative* est aussi un terme-clé. Dans le corpus MÉTIER, le nombre d'occurrences de chaque terme-clé est variable. Certains termes sont très fréquents comme *formation* ou *Association foncière urbaine* avec plus de 500 occurrences ; d'autres termes sont fréquents comme *prix*, *publicité* ou encore *accord-cadre* avec au moins 200 occurrences. Enfin, des termes peu fréquents existent aussi comme *survie*, *contamination*, *réception* ou encore *équipement* avec moins de 50 occurrences. Si nous prenons en considération seulement les articles, la TABLE *1* décrit le nombre d'annotations des experts pour un regroupement d'articles par domaine. Pour ces articles, il est à noter que l'ensemble des termes-clés d'un domaine donné fait référence à un thésaurus de ce domaine.

Domaine	**Nombre de termes-clés**	**Nombre d'articles**	**Nombre d'annotations**
État civil et Cimetières	642	2 767	2 169
Élections	108	152	150
Commande publique	876	1 354	1 201
Urbanisme	327	1 357	554
Comptabilité et finances locales	981	1 971	1 957
Ressources humaines territoriales	293	361	122
Justice	1 447	3 980	870
Santé	491	896	830

TABLE 1 : Taille du vocabulaire et nombre d'annotations des experts

Nous avons récupéré l'ensemble des termes-clés représentant les annotations des experts (à la fois les thésaurus mais aussi des tags utilisés dans les ouvrages). Ces termes-clés ont été décrits avec plusieurs formes fléchies et parfois avec des informations additionnelles non pertinentes comme les déterminants. Par exemple, *des frais* ou *associations syndicales autorisées*. Cela est tout à fait normal puisqu'il n'y avait pas une ressource lexicale de référence au moment de l'annotation, raison pour laquelle tous les experts n'annotent pas de la même façon. Un prétraitement est donc nécessaire. Nous utilisons le parseur Stanza (Qi et al., 2020) pour éliminer les premiers mots outils. Par exemple, le terme *des frais* est transformé en *frais*. Toutefois, *frais* est ambiguë syntaxiquement (nom/adjectif). Afin de lever cette ambiguïté, nous utilisons Stanza pour fournir la classe grammaticale en contexte. Ainsi, chaque terme-clé est associé avec (1) sa classe grammaticale et (2) sa forme lemmatisée prétraitée. Cette combinaison représente l'identifiant d'un terme-clé. Pour les multi-mots, la classe du gouverneur est celle qui est attribuée au terme-clé.

4 Méthodologie

Cette section décrit l'architecture de notre approche et le modèle d'apprentissage que nous avons développé. Nous répondons au problème d'extraction de relations en plusieurs étapes : (1) Quelle forme fléchie est la plus adéquate pour un terme-clé donné ? (2) Quel est le réseau lexico-sémantique français à utiliser pour la validation des relations pouvant être extraites ? (3) Comment les vecteurs de relations sont construits à partir d'un modèle Word2Vec ? et (4) Quel modèle de classification est développé pour juger la pertinence des relations d'hyperonymie et d'hyponymie ? Dans ce qui suit, nous faisons référence à l'hyperonymie par la relation *r_isa* et à l'hyponymie par la relation *r_hypo.*

Dans la première étape, les termes-clés d'un même identifiant vont être unifiés. Nous souhaitons donner une même forme aux termes dits équivalents mais de forme fléchie différente. Par exemple, *personne âgée* et *personnes âgées* font référence au même terme-clé. Afin de choisir le bon représentant, nous avons fait le choix de prendre la forme fléchie ayant le plus grand nombre d'occurrences dans le corpus MÉTIER. Pour cela, une analyse statistique est effectuée sur tout le corpus afin de calculer le nombre d'occurrences de chaque forme fléchie pour chaque terme-clé annoté par les experts. Ainsi, la forme la plus fréquente représente le substitut pouvant faire référence à un terme-clé donné. Pour les termes-clés dits simples (par exemple, *réception*, *équipement*, *marché*, etc.), nous privilégions la forme singulière à la forme plurielle. Cela se justifie par le fait d'avoir une forme standard pouvant être retrouvée facilement dans des dictionnaires. Afin de satisfaire ce besoin, nous utilisons en plus de Stanza la ressource Lexique3 (New et al., 2007) pouvant avoir une priorité plus élevée dans notre processus.

Pour la seconde étape, notre choix s'est porté sur le réseau lexico-sémantique JeuxDeMots[1] (Lafourcade, 2007). En effet, à ce jour, JeuxDeMots est le plus grand réseau français librement disponible avec 14 millions de nœuds et environ 320 millions de relations. Il permet d'avoir le sens commun et les sens métiers des termes polysémiques français. Son utilisation nous permet d'extraire un ensemble de relations de type *r_isa* et *r_hypo* et cela à partir de la liste des termes-clés substituts du corpus MÉTIER. Dans cet article, nous nous intéressons seulement à ces deux types de relations. L'utilisation de JeuxDeMots nous a permis de récupérer un ensemble de 110 118 relations d'hyperonymes. En inversant ces relations, cela nous permet d'avoir un même nombre de relations d'hyponymes. Les paires extraites de JeuxDeMots constituent un jeu d'apprentissage et d'évaluation.

Pour la troisième étape, nous entraînons des plongements lexicaux sur le corpus MÉTIER afin d'obtenir des représentations vectorielles continues de termes-clés. Pour cela, nous faisons tout d'abord une substitution lexicale dans le corpus de tout terme-clé par son référant (forme fléchie la plus fréquente ou représentation dans Lexique3 pour les mots simples). Ensuite, le modèle Word2Vec (Mikolov et al., 2013) avec une architecture CBOW (*Continuous Bag-of-Words*) est utilisé pour l'entraînement. Nous avons utilisé Gensim[2] pour satisfaire cet entraînement (avec un paramétrage par défaut). Par ailleurs, nous avons privilégié l'utilisation de Word2Vec à la place des modèles d'entraînement de plongements lexicaux contextualisés comme BERT (Devlin et al., 2019). Cela se justifie par le fait de ne pas se limiter seulement à des contextes où le terme-source et le terme-cible d'une relation donnée doivent co-occurrer dans une même phrase ou un même paragraphe. Le fonctionnement de BERT est limité à 512 mots en contexte et les termes reliés par hyperonymie ne sont pas forcément en simultané dans un même paragraphe. Il est donc intéressant d'englober une grande quantité de textes avec Word2Vec. Par ailleurs, et pour un principe un peu différent, nous

[1] http://www.jeuxdemots.org/rezo.php

[2] https://radimrehurek.com/gensim/models/word2vec.html

n'avons pas utilisé FastText (Bojanowski et al., 2017). Le concept de *n*-grammes tel qu'il est décrit par cette bibliothèque d'apprentissage pose un problème de compositionnalité des mots. Par exemple, le vecteur associé au terme *fiche de paie* par FastText est le vecteur moyen des mots qui le composent, à savoir : *fiche* et *paie*. Avec Word2Vec, nous pouvons forcer la compréhension des suites de mots comme un élément à part entière (i.e. *fiche_de_paie*). Ainsi, dans notre travail, les plongements lexicaux représentant nos termes-clés sont appris en les considérant comme des entités et non comme une moyenne des mots qui les composent. Pour cela, nous utilisons les phrases du module *gensim*[3].

La suite de notre processus d'entraînement avec Word2Vec consiste à construire des vecteurs de relations à partir des vecteurs de termes-clés. Pour cela, nous proposons de calculer un nouveau vecteur par soustraction des vecteurs en entrée. Concrètement, si V_1 est le vecteur du terme-clé *marché public* et V_2 est le vecteur du terme-clé *marché* alors (V_1–V_2) est le vecteur de la relation (*marché public*, *r_isa*, *marché*). Cette représentation vectorielle répond à la contrainte de non-symétrie des relations *r_isa* et *r_hypo*. Ces plongements lexicaux sont le point d'entrée pour la classification.

En dernière étape, nous avons créé plusieurs classifieurs binaires : de l'utilisation des arbres de décision et des méthodes ensemblistes vers l'utilisation des machines à vecteur de support (*Support Vector Machine – SVM*). Pour cela, nous avons utilisé la bibliothèque Scikit-learn (Buitinck et al., 2013). Chaque classifieur permet de prédire si une paire de termes fait référence à une association soit entre un terme-cible et un terme générique, soit entre un terme-cible et un terme spécifique. Les vecteurs *embeddings* de relations sont fournis aux classifieurs. Nous avons à disposition un ensemble de 220 236 paires de relations (moitié *r_isa*/moitié *r_hypo*). La base d'exemples est équilibrée puisqu'elle a été construite à partir de la relation *r_isa*.

Le principe de notre approche est d'apprendre des relations sémantiques tirées de JeuxDeMots et de les évaluer. L'objectif, par la suite, étant d'utiliser ces classifieurs binaires pour prédire de potentielles relations sémantiques non reconnues à l'heure actuelle par JeuxDeMots. En effet, même si JeuxDeMots représente l'un des plus grands réseaux lexico-sémantiques du français, la couverture des relations sémantiques contenant des termes provenant de domaines métier reste à améliorer. Ainsi, le développement de classifieurs pour la prédiction de nouvelles relations peut être vu comme un axe d'enrichissement de telles ressources. Dans la section qui suit, nous discutons des résultats obtenus.

5 Résultats et discussion

Nous avons appliqué nos classifieurs binaires sur un jeu de données équilibré de 220 236 paires de relations (110 118 relations *r_isa* / 110 118 relations *r_hypo)*. Les classifieurs utilisés ont été instanciés avec les paramètres par défaut et une identification des meilleurs paramètres à l'aide de *GridSearchCV*[4] n'a pas eu lieu. Afin d'obtenir des résultats fiables, nous utilisons le principe de la validation croisée (*Cross-validation*). Une fonction est disponible dans Scikit-learn[5]. Afin d'évaluer la qualité de nos classifieurs, nous utilisons deux mesures d'évaluations, à savoir : (1) le taux d'exactitude (*accuracy rate*) et (2) F-mesure. Ces deux mesures ont été indiquées directement dans un paramètre d'entrée à la fonction de validation croisée de Scikit-learn. Les résultats obtenus sont présentés dans le tableau 2.

[3] https://radimrehurek.com/gensim/models/phrases.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

Modèle	F-mesure	Taux d'exactitude
LinearSVC	0,73	0,85
RidgeClassifier	0,72	0,79
KNeighborsClassifier	0,82	0,91
RandomForestClassifier	0,69	0,76
DecisionTreeClassifier	0,67	0,73
LogisticRegression	0,71	0,83
SupportVectorClassification	**0,83**	**0,92**

TABLE 2 : Résultats d'évaluation par application de différents classifieurs

Nous constatons des résultats entre 67 % et 83 % pour la F-mesure, ce qui nous montre que cette voie est prometteuse. Nous constatons aussi que les moins bons résultats sont obtenus sur les classifieurs de type arbres de décision. En effet, les *embeddings* qui sont entraînés présentent en sortie un certain nombre de dimensions. Les arbres de décisions, eux, extraient les dimensions qu'ils considèrent comme importantes. Toutes les dimensions ne sont donc pas traitées. Par ailleurs, nous avons de bons résultats avec un autre type de classifieurs, à savoir : les k plus proches voisins (avec $k = 3$). La famille des machines à vecteurs de support (SVM) se servent de vecteurs afin de discriminer une classe. Elles se sont avérées plus adaptées à l'utilisation des plongements lexicaux. De plus, nous pouvons obtenir encore de meilleurs résultats en utilisant *GridSearchCV* pour fixer les meilleurs hyperparamètres.

6 Conclusion et perspectives

Dans cet article, nous avons proposé une approche d'extraction de relations sémantiques basée sur des modèles de classification. Deux relations ont été traitées, à savoir : l'hyperonymie et l'hyponymie. Des plongements lexicaux ont été entraînés sur un corpus de spécialité métier décrivant 8 domaines. À partir de ces plongements lexicaux, des vecteurs de relations ont été créés et fournis à plusieurs classifieurs binaires pour prédire la relation pouvant lier deux termes donnés. L'originalité de ce travail réside dans l'apprentissage des vecteurs de relations guidé par une ressource lexico-sémantique, à savoir JeuxDeMots. Les résultats obtenus sont prometteurs : une F-mesure de 83 % et un taux d'exactitude de 92 % par utilisation de l'algorithme de classification SVC (*Support Vector Classification*) faisant référence à la famille des machines à vecteur de support (SVM).

Trois perspectives s'ouvrent à nous à la suite de ce travail : (1) nous envisageons d'intégrer une couche de déduction des relations par transitivité. Par exemple, si ($terme_A$ *r_isa* $terme_B$) et ($terme_B$ *r_isa* $terme_C$) alors ($terme_A$ *r_isa* $terme_C$). Cela permettrait, d'une part, d'améliorer le taux de performance de nos classifieurs en augmentant la base d'apprentissage, et d'autre part, de structurer les données par une extraction de la hiérarchie de termes pouvant être obtenue ; (2) nous envisageons d'extraire des relations comme précédemment énoncé mais pour un seul domaine de spécialité à la fois. Cela permettrait, par exemple, d'orienter la construction d'une ontologie à un seul domaine prédéfini. Pour cela, JeuxDeMots peut être utilisé en sélectionnant les termes liés à la relation *Domaine* ; et (3) notre étude peut être étendue à d'autres relations, par exemple la synonymie ou l'antonymie qui peuvent fonctionner aussi par paires de termes. Pour de telles relations symétriques, la modification du vecteur de relation ($V_{termeA} - V_{termeB}$) vers ($|V_{termeA} - V_{termeB}|$) est essentielle. Toutefois, le non-respect de la symétrie peut être un indicateur de problèmes dans l'apprentissage et/ou la polysémie (sens commun/métier).

Références

ASIM M. N., WASIM M., KHAN M. U. G., MAHMOUD W. & ABBASI H. M. (2018). A survey of ontology learning techniques and applications. *Database*, 2018: article ID bay101. DOI: 10.1093/database /bay101.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, **5**, p. 135–146.

BORDEA G., BUITELAAR P., FARALLI S. & NAVIGLI R. (2015). SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015@NAACL-HLT)*. **452**(465), p. 902–910.

BRIN S. (1998). Extracting Patterns and Relations from the World Wide Web. In *International Workshop on The World Wide Web and Databases*, Springer, p. 172–183.

BUITELAAR P., CIMIANO P. & MAGNINI B. (2005). Ontology learning from text: an overview. In *Ontology Learning from Text: Methods, Evaluation and Applications,* IOS Press, **123**, p. 3–12.

BUITINCK L., LOUPPE G., BLONDEL M., PEDREGOSA F., MUELLER A., GRISEL O., NICULAE V., PRETTENHOFER P., GRAMFORT A., GROBLER J., LAYTON R., VANDERPLAS J., JOLY A., HOLT B. & VAROQUAUX G. (2013). API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, p. 108–122.

BUNESCU R. & MOONEY R. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, p. 724–731.

CARTIER E. (2015). Extraction automatique de relations sémantiques dans les définitions : approche hybride, construction d'un corpus de relations sémantiques pour le français. *Traitement Automatique des Langues Naturelles (TALN),* Caen, France. HAL : halshs-01412736.

CULLEN J. & BRYMAN A. (1988). The Knowledge Acquisition Bottleneck: Time for Reassessment? *Expert Systems*, **5**, p. 216–225.

DEB C. K., MARWAHA S., ARORA A. & DAS M. (2018). A Framework for Ontology Learning from Taxonomic Data. In *Big Data Analytics, Springer*, p. 29–37.

DENDIEN J. & PIERREL J.-M. (2003). Le Trésor de la Langue Française Informatisé : un exemple d'informatisation d'un dictionnaire de langue de référence. *Traitement Automatique des Langues, HERMÈS.*

DEVLIN J., CHANG M., LEE K. & TOUTANOVA K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-Human Language Technology*.

ETZIONI O., CAFARELLA M. J., DOWNEY D., POPESCU A., SHAKED T., SODERLAND S., WELD D. S. & YATES A. (2004). Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. *American Association for Artificial Intelligence (AAAI 2004),* p. 391–398.

GRANADA R., VIEIRA R., TROJAHN C. & AUSSENAC-GILLES N. (2018). Evaluating the Complementarity of Taxonomic Relation Extraction Methods Across Different Languages. arXiv: 1811.03245v1.

HASHIMOTO K., STENETORP P., MIWA M. & TSURUOKA Y. (2015). Task-Oriented Learning of Word Embeddings for Semantic Relation Classification. *Computational Natural Language Learning (CoNLL)*.

HEARST M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, **2**, *COLING '92*, p. 539–545, Stroudsburg, PA, USA.

KONYS A. (2019). Knowledge Repository of Ontology Learning Tools from Text. In *the 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, p. 1614–1628.

LAFOURCADE M. (2007). Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *SNLP'07: 7th International Symposium on Natural Language Processing*, Dec 2007, Pattaya, Chonburi, Thailand. HAL : lirmm-00200883.

MALLART C., NOUY M. L., GRAVIER G. & SEBILLOT P. (2020). Relation, es-tu là ? Détection de relations par LSTM pour améliorer l'extraction de relations. *JEP-TALN-RECITAL*.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations*, p. 1–12.

MINTZ M., BILLS S., SNOW R., & JURAKSKY D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, p. 1003–1011.

MORIN E. & JACQUEMIN C. (2004). Automatic Acquisition and Expansion of Hypernym Links. *Computers and the Humanities*, 38, p. 363–396.

NEW B., BRYSBAERT M., VÉRONIS J. & PALLIER C. (2007). The use of film subtitles to estimate word frequencies. *Applied Psycholinguistics*, p. 661–677.

PANCHENKO A., FARALLI S., RUPPERT E., REMUS S., NAETS H., FAIRON C., PONZETTO S. P. & BIEMANN C. (2016). TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling. In *Proceedings of SemEval-2016 (SemEval@NAACL-HLT)*, p. 1320–1327.

PANCHENKO A., NAETS H., BROUWERS L. & FAIRON C. (2013). Recherche et visualisation de mots sémantiquement liés. *TALN-RÉCITAL 2013*, p. 747–754.

PANTEL P. & PENNACCHIOTTI M. (2006). Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, p. 113–120.

QI P., ZHANG Y., ZHANG Y., BOLTON J. & MANNING C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *System Demonstrations, Association for Computational Linguistics (ACL)*.

SNOW R., JURAFSKY D. & NG A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems (NIPS 2004)*.

WANG C., HE X. & ZHOU A. (2017). A Short Survey on Taxonomy Learning from Text Corpora: Issues, Resources and Recent Advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP),* p. 1201–1214.

XUE L., QING S. & PENGZHOU Z. (2018). Relation Extraction Based on Deep Learning. *The 17th International Conference on Computer and Information Science (ICIS)*, p. 687–691, DOI: 10.1109/ICIS.2018.8466437.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Formalisation de la relation entre les verbes imperfectifs et perfectifs en ukrainien

Olena Saint-Joanis[1, 2], Max Silberztein[1]
(1) ELLIADD, Université Bourgogne Franche-Comté, 30-32 rue Mégevand, 25030 Besançon, France
(2) CREE, INALCO, 2 Rue de Lille, 75007 Paris, France
max.silberztein@univ-fcomte.fr, olena.saint-joanis@inalco.fr

RESUME

Dans la tradition linguistique slave, les formes perfectives et imperfectives des verbes sont traditionnellement inscrites séparément dans les dictionnaires. Cependant, il existe de forts liens morphologiques et sémantiques entre les deux formes verbales. Nous présentons une formalisation qui nous a permis de lier les deux formes. Nous avons construit un dictionnaire électronique qui contient plus de 13 000 entrées verbales associées à plus de 300 paradigmes morphologiques, qui peut être utilisé pour automatiquement lemmatiser les formes verbales dans les textes ukrainiens et relier les formes perfectives et imperfectives.

ABSTRACT

In the Slavic linguistic tradition, perfective and imperfective forms of verbs are traditionally entered independently in dictionaries. However, there are strong morphological and semantic connections between the two forms. We present a formal framework that has allowed us to link the two forms. We have constructed an electronic dictionary that contains over 13,000 entries associated with over 300 morphological paradigms that can be used to automatically lemmatize verbal forms in Ukrainian texts and link the perfective and imperfective forms.

MOTS-CLES : formalisation, couple aspectuel, ukrainien, verbe, imperfectif, perfectif
KEYWORDS: formalization, aspectual pairs, Ukrainian, verb, imperfective, perfective

1 Introduction

Les dictionnaires des langues slaves traditionnels présentent les verbes sous leur aspect perfectif et imperfectif comme des entrées indépendantes. Dans le cadre de la grammaire transformationnelle de Zellig Harris[1], on aimerait pouvoir relier les deux phrases suivantes :

1. ІВАН *РОБИВ* ВПРАВУ ДВІ ГОДИНИ. [Ivan *a fait* = *travaillé sur* l'exercice pendant deux heures].
2. ІВАН *ЗРОБИВ* ВПРАВУ ЗА ДВІ ГОДИНИ. [Ivan *a fait* = fini l'exercice en deux heures].

Etablir le lien entre ces deux phrases signifie qu'on ne traite pas les deux formes *РОБИВ* et *ЗРОБИВ* comme deux objets linguistiques indépendants, mais qu'elles représentent plutôt deux facettes d'une même entité, ce qui implique d'une part de représenter cette entité par une unique

[1] Cf. (Harris 1988) et (Harris 1991).

entrée lexicale, et d'autre part de construire une correspondance bi-univoque qui permette à loisir de calculer une forme à partir de l'autre, et réciproquement.

Pour formaliser ce lien, nous avons utilisé la plateforme de développement NooJ[2]. NooJ permet d'une part de construire des dictionnaires électroniques[3], d'autre part d'associer les entrées de ces dictionnaires avec des grammaires morphologiques pour formaliser trois types de phénomènes : les flexions (par ex. la conjugaison des verbes), les dérivations (par ex. l'opération de nominalisation) et les agglutinations (par ex. la préfixation en *re-*). Une caractéristique des ressources linguistiques de NooJ est qu'elles sont réversibles : ainsi, une même règle morphologique peut être utilisée à la fois pour dériver le verbe *manifester* en *manifestant*, mais aussi pour retrouver le verbe à partir de la forme nominale. Cette caractéristique est essentielle si l'on veut implémenter une grammaire transformationnelle capable de relier les phrases perfectives et imperfectives dans les deux sens. Un second avantage est que le logiciel NooJ peut aussi être utilisé comme outil de linguistique de corpus : dès que l'on a construit une ressource linguistique, on peut la tester en l'appliquant à n'importe quel corpus (non préalablement étiqueté).

2 Cadre théorique

La notion d'aspect perfectif ou imperfectif a été utilisée pour la première fois pour la langue tchèque en 1603 par Vavřinec Benedikt Nedožerský dans sa *Grammaticae Bohemicae* pour distinguer des verbes simples de verbes dérivés (A. Mazon). Ses successeurs ont développé et étendu cette notion aux autres langues slaves. L'aspect se base sur les diverses façons de concevoir le procès exprimé par le verbe dans le temps (J. Holt) et s'exprime par des formes imperfectives (IPF) et perfectives (PF). Les formes imperfectives focalisent l'attention sur le processus même de l'action, tandis que les formes perfectives expriment le franchissement des limites. La plupart des verbes imperfectifs et perfectifs forment des couples aspectuels (IPF/PF) qui expriment une même action ou état, mais d'un point de vue différent (durée, répétition, accomplissement, achèvement...).

Il existe deux écoles de grammairiens.

- La première[4] considère qu'une partie des couples est composée de deux formes aspectuelles du même verbe, c'est-à-dire sémantiquement identiques mais ayant des formes morphologiques différentes. Ajoutons qu'il y a deux types de formes aspectuelles : « préfixales » et les « suffixales ». Les « préfixales » ont le même radical mais se différencient par les préfixes perfectivants tandis que les « suffixales » ont le même radical et le même préfixe et se différencient par les suffixes imperfectivants.

---

[2] NooJ est une plateforme de développement linguistique gratuite et open source, cf. http://www.nooj-associartion.org. Les cadres théorique et méthodologique de NooJ sont décrits par (Silberztein, 2015).

[3] Les dictionnaires électroniques ont des contenus et des formats essentiellement différents de ceux des dictionnaires éditoriaux. Par exemple, les dictionnaires électroniques ne contiennent pas d'information de type encyclopédique ; inversement ils recensent de façon très détaillée et systématique les propriétés orthographiques, morphologiques, syntaxiques et/ou sémantiques des entrées lexicales décrites.

[4] Cf. par exemple (Vinogradov 1986), (Vyhovanets, 2004), (Rousanivskiy, 2001), (Plušč, 2010)

Prenons l'exemple du couple *РОБИТИ* (IPF) /***З****РОБИТИ* (PF) [*faire*]. La forme imperfective n'a ni suffixe, ni préfixe, tandis que la forme perfective est composée du radical *РОБИТИ* et du préfixe perfectivant ***З-***.

- La seconde[5] traite le couple aspectuel comme une opposition entre deux verbes différents qui ont une sémantique proche et dont le deuxième est dérivé du premier grâce à un préfixe ou un suffixe de façon idiosyncratique.

Dans notre exemple ІВАН *ЗРОБИВ* ВПРАВУ ЗА ДВІ ГОДИНИ [Ivan *a fait* l'exercice en deux heures], le verbe ***З****РОБИТИ* (PF) serait ainsi dérivé du verbe *РОБИТИ* (IPF) par ajout du préfixe ***З-*** qui ajoute une information sémantique supplémentaire : un résultat abouti dans une limite temporelle. Selon cette approche, un même verbe peut former dans des contextes différents des couples aspectuels occasionnels, sémantiquement irréguliers. Citons un exemple pour illustrer cela :

ІВАН *РОБИВ* ВПРАВУ ДВІ ГОДИНИ, АЛЕ НЕ *ДОРОБИВ* ЇЇ. [Ivan *a fait* l'exercice pendant deux heures, mais ne *l'a pas fait jusqu'à la fin*]. Ici le couple aspectuel est *РОБИТИ* (IPF) / ***ДО****РОБИТИ* (PF). Le verbe perfectif est formé par ajout du préfixe ***ДО-*** au verbe [*faire*] et porte l'information sur la limite [*faire jusqu'à la fin*].

En poussant cette approche, un couple aspectuel peut très bien être formé à partir de deux verbes morphologiquement éloignés, comme par exemple *ГОВОРИТИ* (IPF) / *СКАЗАТИ* (PF) [*dire*] ou *БРАТИ* (IPF) / *ВЗЯТИ* (PF) [*prendre*] ; inversement, certains verbes (bi-aspectuels) peuvent se comporter dans une phrase comme perfectifs, et dans une autre comme imperfectifs.

En ukrainien, il existe 9 préfixes perfectivants[6] que (Guiraud-Weber 1987) considère « dé-sémantisés » qui ne modifient pas le sens du lexème verbal. Il existe en tout 37 préfixes verbaux simples et 3 doubles[7]. Un verbe imperfectif simple (radical + terminaison) peut être associé à plusieurs préfixes qui portent des variations sémantiques. Les variantes peuvent être traduites par un même verbe dans une langue étrangère non slave, comme dans l'exemple cité plus haut, ou alors par des verbes différents.

Par exemple le verbe ***ПЕРЕ****РОБИТИ* (PF), créé par ajout du préfixe ***ПЕРЕ-*** au verbe [faire] se traduit par [refaire], ou encore le verbe ***ПІД****РОБИТИ* (PF), créé par ajout du préfixe ***ПІД-*** au verbe [faire] se traduit par [avoir un emploi]. Ces variantes peuvent être associées à leur tour à d'autres verbes pour former des couples aspectuels : ***ПЕРЕ****РОБИТИ* (PF)/***ПЕРЕ****РОБ****ЛЮВА****ТИ* (IPF) [refaire], ***ПІД****РОБИТИ* (PF)/***ПІД****РОБ****ЛЮВА****ТИ* (IPF) [avoir un emploi]. Bien entendu, les verbes ***ПЕРЕ****РОБ****ЛЮВА****ТИ* (IPF) [refaire] et ***З****РОБИТИ* (PF) [faire] ne constitueront jamais un couple aspectuel à cause de leur éloignement sémantique ; de même, les paires de verbes *РОБИТИ* (IPF) [faire]/***ПЕРЕ****РОБИТИ* (PF) [refaire] ne trouveront jamais un contexte leur permettant de former un couple aspectuel. De ce fait, nous parlerons dans ces cas de couples aspectuels dominants, qu'il conviendra de lier dans le cadre de la grammaire transformationnelle.

---

[5] Cf. par exemple (Karcevski, 1927), (Maslov 1984), (Włodarczyk ,2001), (Gwizdecka, 2009)

[6] Gorpynyč (2004) donne 7 préfixes plus 2 variantes du préfixes з- : зі-, с-

[7] Nous prenons en compte toutes les variantes

3 Implémentation dans NooJ

Dans un dictionnaire NooJ, les entrées lexicales sont associées à une catégorie morpho-syntaxique (ex. « V » pour *verbe*) et à un certain nombre de paradigmes morphologiques de deux types :

- les paradigmes de type « FLX » permettent d'associer l'entrée lexicale à un ensemble de formes de catégorie identique à l'entrée lexicale ; dans les langues romanes, on utilise typiquement des paradigmes « FLX » pour fléchir des mots, par ex. *manifester* ⟶ *manifesteront* ;
- les paradigmes de type « DRV » permettent d'associer l'entrée lexicale à une forme dont la catégorie peut être différente ; dans les langues romanes, on utilise typiquement des paradigmes « DRV » pour dériver des mots, par ex. nominaliser un verbe (*manifester* ⟶ *manifestant*).

La distinction entre flexion et dérivation n'a bien entendu pas de sens pour l'ordinateur[8], et des problèmes de limite se posent souvent. Par exemple, pour le français, (Silberztein, 2015a) a choisi d'intégrer à la conjugaison (paradigme « FLX ») les formes participiales des verbes (*manger* ⟶ *mangée*) alors qu'il y a de bonnes raisons pour considérer ces formes comme adjectivales (comme *mangeable*). Inversement, il a choisi de décrire les préfixations (ex. *manger* ⟶ *remanger*) comme des phénomènes dérivationnels (paradigme « DRV »), même si les formes produites restent des formes verbales fortement associées au lemme initial.

Les paradigmes FLX et DRV sont décrits par des grammaires hors-contexte à transduction[9]. Par exemple, le paradigme FLX= ВИКЛАЛАДАЧ défini par la règle de grammaire suivante :

ВИКЛАЛАДАЧ = <E>/+singulier+masculin | ка/+singulier+féminin | і/+pluriel+masculin | ки/+pluriel+féminin[10] ;

peut être associé à l'entrée lexicale *ВИКЛАЛАДАЧ* [enseignant] :

виклаладач,NOM+FLX=ВИКЛАЛАДАЧ

pour l'associer aux quatre formes *ВИКЛАЛАДАЧ* [enseignant], *ВИКЛАЛАДАЧКА* [enseignante], *ВИКЛАЛАДАЧІ* [enseignants] et *ВИКЛАЛАДАЧКИ* [enseignant]. De même, le paradigme dérivationnel DRV = PROFESSION_Ч défini par la règle suivante :

PROFESSION_Ч = <B2>ч/NOM

permet de décrire la nominalisation de l'entrée verbale *ВИКЛАДАТИ* [enseigner]. En association les deux règles précédentes, on peut alors relier l'entrée verbale *ВИКЛАДАТИ* [enseigner] aux

---

[8] Pour des raisons historiques, le logiciel NooJ avait été au départ conçu pour formaliser les langues romanes, et les codes NooJ « FLX » (flexion) et « DRV » (dérivation) reflètent donc cette histoire, mais il ne faut pas confondre ces codes avec un concept linguistique, surtout lorsqu'on décrit des langues non-romanes.

[9] Le logiciel NooJ permet d'écrire quatre types de grammaires : les grammaires rationnelles, les grammaires algébriques ou hors-contexte, les grammaires contextuelles et les grammaires non restreintes.

[10] NooJ contient une douzaine d'opérateurs morphologiques de base, et chaque langue a ses propres opérateurs. L'opérateur <B> permet d'effacer une lettre ; le symbole <E> correspond au suffixe vide, etc.

quatre formes nominales *ВИКЛАЛАДАЧ* [enseignant], *ВИКЛАЛАДАЧКА* [enseignante], *ВИКЛАЛАДАЧІ* [enseignants], *ВИКЛАЛАДАЧКИ* [enseignantes] :

викладати,VERBE+DRV=PROFESSION_Ч:ВИКЛАЛАДАЧ

Pour relier les formes perfective et imperfective dans les langues slaves, a priori deux solutions pourraient donc être choisies :

1. Inscrire en entrée du dictionnaire la forme imperfective de chaque verbe, puis décrire les formes perfectives à l'aide de paradigmes « FLX » (comme une simple conjugaison) :

verbe_imperfectif,VERBE+FLX=*conjugaison_imperfective*+FLX=*conjugaison_perfective*

2. Inscrire en entrée du dictionnaire la forme imperfective de chaque verbe, puis la lier à la forme perfective correspondante grâce à un paradigme « DRV » (comme une dérivation), puis décrire les formes conjuguées de celle-ci :

verbe_imperfectif,VERBE+FLX=*conjugaison_imperfective*+DRV=*perfectivation:conjugaison_perfective*

Nous écartons la première solution « FLX », car elle conduirait à produire un très grand nombre de paradigmes flexionnels (puisqu'il faudra combiner les paradigmes flexionnels, avec chacun des préfixes perfectivants potentiels). Nous avons donc choisi la deuxième solution, ce qui nous a conduit à formaliser 186 paradigmes flexionnels et 136 paradigmes dérivationnels.

4 Le dictionnaire ukrainien

Pour construire notre dictionnaire, nous avons utilisé le dictionnaire électronique Open Source version 2.9.1 (Polyakov & Rysin, 2016) qui contient une liste de 20 639 verbes non fléchis. Nous avons retiré de cette liste les variantes en ***В***- de verbes qui commencent en ***У***-, comme par exemple pour le verbe ***У****ЧИТИ*/***В****ЧИТИ* [*apprendre*], ainsi que les variantes postfixées en ***–СЯ***, comme par exemple *МИТИСЯ* [*se laver*], pour ne garder que les entrées sans affixe, ex. *МИТИ* [*laver*]. Les préfixations et suffixations seront en effet traitées comme dérivations. Après ce filtrage, il nous reste 11 545 verbes perfectifs et 8 022 verbes imperfectifs. Parmi ces derniers, 442 verbes sont bi-aspectuels, 1 502 verbes sont toujours imperfectifs, 6 078 forment un couple aspectuel avec un verbe perfectif, et 321 verbes forment un coupe aspectuels avec plusieurs verbes perfectifs. Parmi les verbes perfectifs, nous avons repéré 5 206 verbes qui ne sont pas liés à une forme imperfective. Nous avons donc constitué un dictionnaire de 13 228 entrées verbales : 8 022 verbes purement imperfectifs + 5 206 verbes purement perfectifs, puis avons décrit les 6 339 formes perfectives de verbes imperfectifs grâce à des règles morphologiques. Voici quatre entrées verbales de notre dictionnaire :

	Entrée de dictionnaire	**Aspect**	**Traduction**
1	робити,VERBE+FLX=ЛЮБИТИ+DRV=3:ЛЮБИТИ_2	IPF ⟶ PF	faire
2	анексувати,VERBE+FLX=РИСУВАТИ+FLX=РИСУВАТИ_2	IPF & PF, bi-aspectuel	annexer
3	любити,VERBE+FLX=ЛЮБИТИ	IPF pur	aimer
4	полюбити,VERBE+FLX=ЛЮБИТИ_2	PF pur	tomber amoureux

TABLE 1 : Entrées verbales de dictionnaire

Dans *la TABLE 1* nous pouvons observer :

1. le verbe imperfectif *РОБИТИ* [*faire*] qui est fléchi selon le modèle de conjugaison imperfectif *ЛЮБИТИ*, puis est relié à l'aide de la dérivation *DRV=3* à la forme perfective *ЗРОБИТИ* qui à son tour est fléchie selon le modèle perfectif *ЛЮБИТИ_2*.
2. le verbe bi-aspectuel *АНЕКСУВАТИ [annexer]* qui se fléchit selon le modèle imperfectif *РИСУВАТИ [dessiner]*, et selon le modèle perfectif *РИСУВАТИ_2*
3. le verbe imperfectif *ЛЮБИТИ* [*aimer*] qui se fléchit selon le modèle *ЛЮБИТИ*
4. le verbe perfectif *ПОЛЮБИТИ* [*tomber amoureux*] qui se fléchit selon le modèle *ЛЮБИТИ_2*

Les langues slaves, et en particulier l'ukrainien, ont une morphologie flexionnelle lourde : chaque forme verbale est en effet associée à un temps, un mode, une personne, un genre et un aspect. La *Table 2* présente les paradigmes flexionnels (FLX) ЛЮБИТИ et ЛЮБИТИ_2 utilisés pour fléchir les entrées lexicales présentées dans la *TABLE 1* :

ЛЮБИТИ =	ЛЮБИТИ_2 =
<E>/Imperfective+Infinitive[11]| <B2>в/Imperfective+Masculine+Singular+Indicative+Past| <B2>ла/Imperfective+Feminine+Singular+Indicative+Past| <B2>ло/Imperfective+Neuter+Singular+Indicative+Past| <B2>ли/Imperfective+Plural+Indicative+Past| <B3>лю/Imperfective+1+Singular+Present| <B3>иш/Imperfective+2+Singular+Present| <B3>ить/Imperfective+3+Singular+Present| <B3>имо/Imperfective+1+Plural+Present| <B3>ите/Imperfective+2+Plural+Present| <B3>лять/Imperfective+3+Plural+Present| <E>му/Imperfective+1+Singular+Future| <E>ме/Imperfective+2+Singular+Future| <E>меш/Imperfective+3+Singular+Future| <E>мемо/Imperfective+1+Plural+Future| <E>мете/Imperfective+2+Plural+Future| <B3>и/Imperfective+2+Singular+Imperative| <B3>імо/Imperfective+1+Plural+Imperative| <B3>ім/Imperfective+1+Plural+Imperative| <B3>іть/Imperfective+2+Plural+Imperative| <B3>лячи/Imperfective+Gerund+Present ;	<E>/Perfective+Infinitive| <B2>в/Perfective+Masculine+Singular+Indicative+Past| <B2>ла/Perfective+Feminine+Singular+Indicative+Past| <B2>ло/Perfective+Neuter+Singular+Indicative+Past| <B2>ли/Perfective+Plural+Indicative+Past| <B3>лю/Perfective+1+Singular+Present| <B3>иш/Perfective+2+Singular+Present| <B3>ить/Perfective+3+Singular+Present| <B3>имо/Perfective+1+Plural+Present| <B3>ите/Perfective+2+Plural+Present| <B3>лять/Perfective+3+Plural+Present| <B3>и/Perfective+2+Singular+Imperative| <B3>імо/Perfective+1+Plural+Imperative| <B3>ім/Perfective+1+Plural+Imperative| <B3>іть/Perfective+2+Plural+Imperative| <B2>вши/Perfective+Gerund+Past ;

TABLE 2 : Deux paradigmes flexionnels

La *TABLE 3* présente quelques paradigmes dérivationnels (DRV) utilisés pour relier des entrées lexicales imperfectives à leur forme perfective :

DRV préfixale	DRV suffixale	DRV prenant en compte l'alternance des voyelles
З=<LW>з/VERBE[12] ; В=<LW>в/VERBE ;	А_И_1=<B3>ити/VERBE ; НУ_1=<B3>нути/VERBE ;	ALT_1=<B3><L><B>е<RW>ти/VERBE ;

TABLE 3 : Exemple de description des dérivations (DRV)

[11] Les propriétés flexionnelles (par ex. «Infinitive») sont elles-mêmes formalisées par une grammaire décrite dans le fichier « properties.def » du module ukrainien de NooJ

[12] Il s'agit de la DRV=3

Chaque entrée verbale de notre dictionnaire contient ainsi une propriété FLX utilisée pour la conjuguer, et potentiellement une paire de propriétés DRV:FLX pour l'associer à sa forme perfective et conjuguer cette dernière.

5 Evaluation et perspectives

Nous avons ainsi construit manuellement un dictionnaire de 13 228 entrées verbales, associées à leurs paradigmes flexionnels et dérivationnels, ce qui représente 281 738 formes verbales fléchies et dérivées. Nous avons appliqué avec NooJ ce dictionnaire à un corpus de 100 textes (extraits de romans de XIX – XXI^{e} siècles, extraits d'articles), qui contient 199 996 formes graphiques. NooJ a alors reconnu 32 755 occurrences verbales, parmi lesquelles nous avons observé 114 faux positifs liés à des ambiguïtés (*i.e.* une précision de plus de 99%) et 311 formes verbales dialectiques non reconnues (*i.e.* un rappel supérieur à 99%).

Nous comptons compléter le dictionnaire en élargissant notre corpus. Notre but est d'ajouter à cette ressource un ensemble de grammaires transformationnelles capables de produire à volonté des phrases imperfectives à partir des phrases perfectives correspondantes, et réciproquement, en utilisant la méthodologie décrite par (Silberztein, 2015b).

Le module ukrainien sera publié en open-source sur le site WEB de NooJ (page "Linguistic Resources"[13])

Références

Gorpynyč, V. O. (2004). Morphologiya ukraïnskoï movy. [Morphologie de la langue ukrainienne]. Akademiya, Kyïv.

Gwizdecka, E (2009). Quelle description pour le préverbe polonais. In: Cognitive Studies, Instytut Slawistyki Polskiej Akademii Nauk, pp. 243-254

Guiraud-Weber, M (1987). Oppositions aspectuelles et sémantisme verbal en russe. In: Revue des études slaves, tome 59, fascicule 3, pp. 585-596.

Harris, Z. (1988). Language and Information, New York, Columbia University Press.

Harris, Z. (1991) A Theory of Language and Information, Oxford, Clarendon Press.

Holt, J. (1943). Études d'aspect. Acta Jutlandica, 15/2, Copengague,Munksgaard.

Karcevski, Serge (2004). Système du verbe russe. Essai de linguistique synchronique, Institut d'Etudes Slaves, Première édition, Praha (1927).

Maslov, J. S. (1984) Očerki po aspectologii [Les études en aspectologie], Leningrad : LGU.

Mazon, André. (1913). La notion morphologique de l'aspect des verbes chez les grammairiens russes. In : Mélanges offerts à M.Emelie Picot, Paris, E.RAkir, pp.343-367.

Plušč, M. Y. (2010). Gramatyka ukraïnskoï movy. Čatyna 1. Morfemika. Slovotvir. Morfologiya. Pidručnyk dlya studentiv filologi čnyh spetsialnostei vyščyh nav čalnyh zakladiv. [Grammaire de la langue ukrainienne. Manuel pour les étudiants en langues dans les établissements d'enseignements supérieur]. Vyšča škola, Kyïv.

Polyakov, M., Rysin, A. (2016). Ukrainian Dictionary Open Source, version 2.9.1 http://extensions.services.openoffice.org/en/project/dict-uk_UA

[13] http://www.nooj-association.org/resources.html

Rousanivskiy, V (2001). Istoriya ukraïnskoï movy. Pidručnyk. [Histoire de la langue ukrainienne. Manuel]. Artek, Kyïv.

Silberztein, M (2015a). La formalisation des langues. L'approche de NooJ. Université de Franche-Comté, ISTE Edition.

Silberztein, Max (2015b). Joe loves Lea: Transformational Analysis of Transitive Sentences. in Formalising Natural Languages with NooJ (9th International NooJ conference, Minsk, Belarus 2015), CCIS Series. Springer Verlag: Heidelberg (2016).

Vinogradov, V. V. (1986) Russkij jazyk. Grammatičeskoe učenie o slove [La langue russe. Etude grammaticale du mot]. Moskva, Vyssšaja škola. Première édition, 1947 ;

Vyhovanets, I. R., Gorodenska, K. G. (2004). Teorretyčna morfologiya ukraïnskoï movy. [Morphologie théorique de la langue ukraininne]. Pulsray, Kyïv.

Włodarczyk, A., Włodarczyk, H. (2001). La Préfixation verbale en polonais. In: Études Cognitives / Studia Kognitywne Nr 4, SOW, Warszawa, p. 93-109.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Intérêt des modèles de caractères pour la détection d'événements

Emanuela Boros[1] Romaric Besançon[2] Olivier Ferret[2] Brigitte Grau[3]
(1) La Rochelle Université, L3i, F-17042 La Rochelle
(2) Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
(3) Université Paris-Saclay, CNRS, LIMSI, ENSIIE, F-91405, Orsay, France
emanuela.boros@univ-lr.fr, romaric.besancon,olivier.ferret@cea.fr,
brigitte.grau@limsi.fr

RÉSUMÉ

Cet article aborde la tâche de détection d'événements, visant à identifier et catégoriser les mentions d'événements dans les textes. Une des difficultés de cette tâche est le problème des mentions d'événements correspondant à des mots mal orthographiés, très spécifiques ou hors vocabulaire. Pour analyser l'impact de leur prise en compte par le biais de modèles de caractères, nous proposons d'intégrer des plongements de caractères, qui peuvent capturer des informations morphologiques et de forme sur les mots, à un modèle convolutif pour la détection d'événements. Plus précisément, nous évaluons deux stratégies pour réaliser une telle intégration et montrons qu'une approche de fusion tardive surpasse à la fois une approche de fusion précoce et des modèles intégrant des informations sur les caractères ou les sous-mots tels que ELMo ou BERT.

ABSTRACT

The interest of character-level models for event detection.

This paper tackles the task of event detection that aims at identifying and categorizing event mentions in texts. One of the difficulties of this task is the problem of event mentions corresponding to misspelled, custom, or out-of-vocabulary words. To analyze the impact of character-level features, we propose to integrate character embeddings, which can capture morphological and shape information about words, to a convolutional model for event detection. More precisely, we evaluate two strategies for performing such integration and show that a late fusion approach outperforms both an early fusion approach and models integrating character or subword information such as ELMo or BERT.

MOTS-CLÉS : Extraction d'information, événements, plongements lexicaux.

KEYWORDS: Information extraction, events, word embeddings.

1 Introduction

Dans cet article, nous nous concentrons plus particulièrement sur la détection d'événements, qui implique l'identification d'instances de types d'événements prédéfinis dans un texte. Ces instances, appelées mentions d'événements ou déclencheurs d'événements, prennent la forme de mots ou d'expressions polylexicales évoquant un type d'événements de façon plus ou moins spécifique. Les approches les plus efficaces pour réaliser cette tâche sont actuellement fondées sur des modèles neuronaux (Chen *et al.*, 2015; Nguyen & Grishman, 2015; Nguyen *et al.*, 2016a,b; Feng *et al.*,

	Tous les mots	Mentions d'événements
entraînement	14 021	931
test	3 553	219
mots inconnus dans le test	930 (26,2%)	66 (30,1%)
mots inconnus avec un mot similaire	825	54

TABLE 1 – Statistiques concernant le vocabulaire des parties entraînement et test du corpus ACE 2005. Mot inconnu : présent dans la partie entraînement mais pas dans la partie test

2016; Zhang *et al.*, 2019; Nguyen & Grishman, 2018) et ont permis en particulier de s'affranchir du problème du choix des traits linguistiques utilisés par les modèles d'apprentissage statistiques. Ces modèles reposent ainsi sur des plongements de mots qui les rendent en principe moins sensibles au problème des déclencheurs non rencontrés lors de l'entraînement puisque ces plongements intègrent une forme de similarité entre les mots.

Toutefois, cette capacité peut varier en fonction des raisons pour lesquelles un déclencheur n'a pas été vu lors de l'entraînement du modèle. Nous illustrons ces différents cas sur la partie anglaise du jeu de données ACE 2005, un corpus standard pour l'évaluation de la détection d'événements dont nous reprenons la subdivision classiquement faite pour cette tâche entre entraînement, validation et test (Ji *et al.*, 2008). Le déclencheur inédit peut ainsi être une variante morphologique d'un déclencheur déjà vu dans l'ensemble des données d'entraînement. Par exemple, *torturing* n'est pas présent dans les données d'entraînement ACE 2005 mais il s'agit d'une variante de *torture*, qui est considéré comme un déclencheur pour le même type d'événements, en l'occurrence *Life.Injury*. En outre, *torturing* est susceptible d'être présent au sein d'un modèle de langue général, auquel cas un modèle de détection d'événements neuronal reposant sur ledit modèle de langue est susceptible de détecter avec succès ce déclencheur.

La situation est différente lorsqu'un déclencheur est absent des données d'entraînement parce qu'il correspond à une version mal orthographiée d'un déclencheur de référence. En effet, dans un tel cas, le modèle de langue ne contient pas nécessairement la version altérée. Par exemple, *aquitted* fait partie du corpus de test ACE 2005 pour référer à un événement *Justice.Sentence* alors que seule *acquitted*, la forme correcte pour ce mot, est présente dans les données d'entraînement. Dans ce cas, il est peu probable que le mot inédit fasse partie du modèle de langue général et, par conséquent, il a peu de chances d'être détecté comme déclencheur d'un événement *Justice.Sentence*. Plus globalement, comme le montre le tableau 1, 30,1 % des déclencheurs du corpus de test ACE 2005 ne sont pas présents dans le corpus d'entraînement mais 88 % de ces déclencheurs absents sont proches (mesurés par un ratio de Levenshtein inférieur à 0,3) de mots du corpus d'entraînement. Le tableau 2 présente des exemples de telles paires de mots. On peut voir qu'en dehors des paires correspondant à des différences de casse (intifada/Intifada) ou relevant de la morphologie flexionnelle (opening/open), certaines paires correspondent à des cas plus complexes relevant de la morphologie dérivationnelle (creating/creation) ou même de relations sémantiques complexes (hacked/attacked) qui ne sont souvent pas capturées par les modèles de plongements de mots.

Différentes stratégies ont été proposées pour traiter le problème de la variabilité lexicale dans les modèles de langue neuronaux. Pour les plongements statiques de mots, fastText (Bojanowski *et al.*, 2017) s'appuie ainsi sur une représentation des mots fondée sur des n-grammes de caractères. Pour les modèles contextuels, ELMo (Peters *et al.*, 2018) exploite une représentation fondée sur les caractères

Type d'événements	Déclencheur inconnu/connu le plus proche
Start-Org	*creating/creation, opening/open, forging/forming, formed/form*
End-Org	*crumbled/crumbling, dismantling/dismantle, dissolved/dissolving*
Transport	*fleeing/flying, deployment/deployed, evacuating/evacuated*
Attack	*intifada/Intifada, smash/smashed, hacked/attacked, wiped/wipe*
End-Position	*retirement/retire, steps/step, previously/previous, formerly/former*

TABLE 2 – Exemples de déclencheurs événementiels de test proches de déclencheurs d'entraînement

construite grâce à un réseau de neurones convolutif (CNN) tandis que BERT (Devlin *et al.*, 2019) adopte une stratégie mixte fondée sur des sous-mots, appelés wordpieces (Luong & Manning, 2016; Kim *et al.*, 2016; Jozefowicz *et al.*, 2016), avec quelques limites sur sa capacité à gérer les entrées bruitées (Sun *et al.*, 2020).

Nos contributions dans cet article sont plus particulièrement axées sur l'intégration de modèles reposant sur le niveau des caractères dans les modèles de détection d'événements pour traiter la question des mots inconnus. Plus précisément, nous montrons qu'un modèle de détection d'événements exploitant une représentation fondée sur les caractères est complémentaire d'un modèle fondé sur les mots et que leur combinaison selon une approche de fusion tardive est plus performante qu'une stratégie de fusion précoce.

2 Modèles

Notre approche s'inscrit dans le droit fil de la plupart des modèles de détection supervisée d'événements en considérant cette tâche comme une forme de classification multiclasse de mots : étant donné une phrase et un ensemble de types d'événements possibles, l'objectif est de prédire pour chacun de ses mots s'il relève ou non d'un de ces types d'événements et le cas échéant, duquel. L'entrée du système est donc un mot cible dans le contexte d'une phrase et sa sortie, un type d'événements ou l'étiquette NONE pour les mots non déclencheurs. Pour étudier l'influence des traits fondés sur les caractères, nous nous appuyons sur le modèle CNN proposé par Nguyen & Grishman (2015). Ce modèle de base est utilisé dans les deux composantes de notre modèle global : le modèle fondé sur les mots, dit modèle CNN mot, et le modèle fondé sur les caractères, dit modèle CNN caractère. Ces deux composantes sont combinées en utilisant soit une approche de fusion précoce, soit une approche de fusion tardive, comme l'illustre la figure 1.

Dans le modèle CNN mot, le contexte d'un mot candidat en tant que mention événementielle est formé par les mots qui l'entourent dans la phrase. Pour tenir compte de la nécessité de gérer des entrées de même dimension, ce contexte prend la forme d'une fenêtre de taille fixe, centrée sur la mention candidate. De ce fait, les parties de phrases dépassant la limite de cette fenêtre sont tronquées tandis qu'un remplissage avec des valeurs nulles (*zero-padding*) est réalisé pour les phrases plus courtes. Au sein de cette fenêtre de contexte, chaque mot est représenté par un plongement de mot et une position relative par rapport à la mention candidate, elle aussi sous la forme d'un plongement. Les plongements de mots et de positions sont concaténés et passés au travers d'une couche de convolution. Plus précisément, un ensemble de filtres convolutifs de tailles différentes sont appliqués et une opération de *max pooling* est appliquée à l'échelle de la fenêtre pour obtenir une

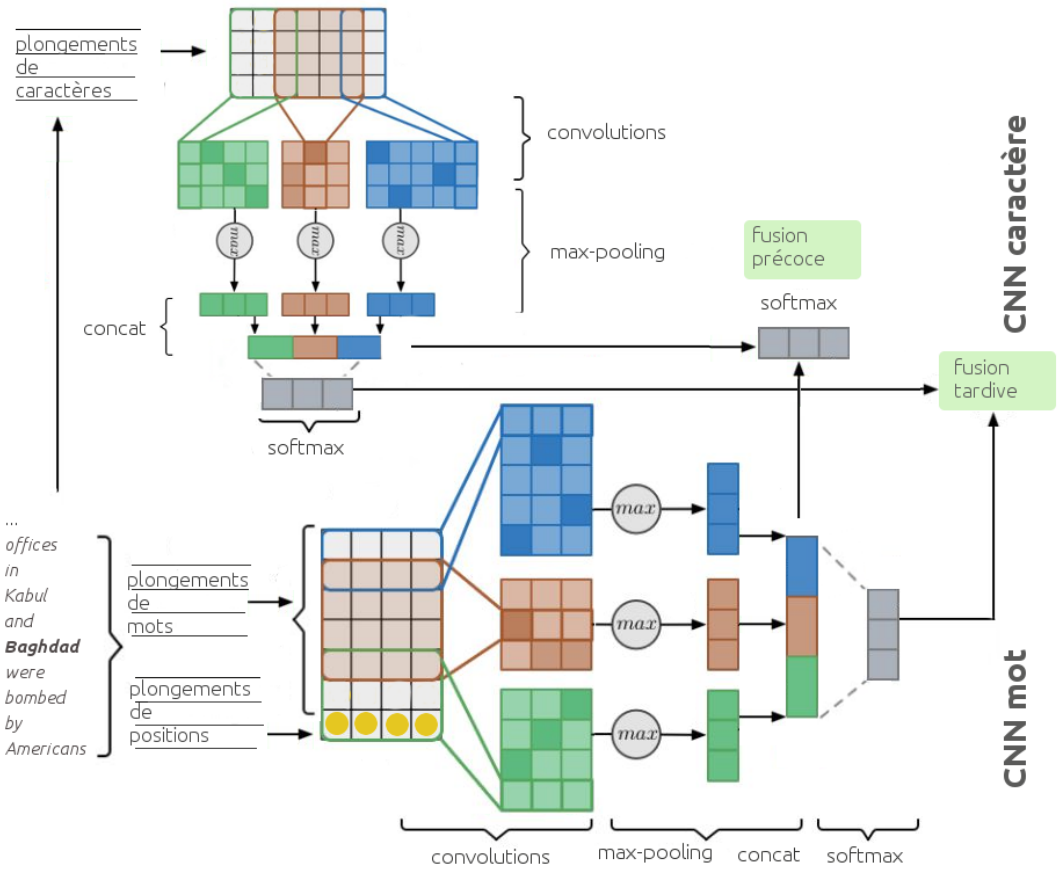


FIGURE 1 – Association d'un modèle fondé sur les mots et d'un modèle fondé sur les caractères

valeur par filtre. Le résultat de ces opérations se voit ensuite appliquer un *softmax* pour réaliser la classification en tant que telle. Le modèle CNN caractère est très proche du modèle CNN mot, avec deux différences principales : les mots sont remplacés par des caractères et il n'y a pas d'information de position associée à chaque caractère. Plus précisément, chaque mention candidate, identifiée sur la base des mots, se voit associer une fenêtre de contexte, comme dans le cas du CNN mot, mais cette fenêtre est dans ce cas déterminée sur la base d'un nombre fixe de caractères et les éléments de base de représentation sont constitués par des plongements de caractères. Les mêmes mécanismes de troncation et de remplissage permettant de considérer des phrases de taille variable et une fenêtre de contexte de taille fixe sont appliqués, mais ici à l'échelle du caractère.

Le premier type d'intégration de ces deux modèles est une fusion précoce, dans laquelle les deux représentations de la séquence d'entrée produites par les CNN de mots et de caractères sont concaténées avant la couche de classification. L'utilisation de ce type d'intégration permet un apprentissage conjoint des paramètres des deux modèles lors de la phase d'entraînement. L'intégration par fusion tardive repose quant à elle sur la combinaison par vote des décisions des deux modèles, qui sont entraînés séparément et apprennent donc des caractéristiques différentes des mentions candidates. La méthode de vote se définit comme suit : si une mention événementielle est détectée par un seul des deux modèles, nous conservons l'étiquette donnée par ce modèle ; sinon, si une mention est détectée par le CNN mot et le CNN caractère ensemble, nous conservons l'étiquette donnée par le CNN caractère. Cette stratégie est motivée par le fait que le modèle CNN mot possède une bonne couverture tandis que le modèle CNN caractère est davantage axé sur la précision.

3 Expérimentations, résultats et discussion

Cadre expérimental Nos expérimentations ont été réalisées sur le corpus ACE 2005. À des fins de comparabilité, nous utilisons le même découpage que les travaux antérieurs (Ji *et al.*, 2008; Liao & Grishman, 2010; Li *et al.*, 2013; Nguyen & Grishman, 2015; Nguyen *et al.*, 2016a), avec 529 documents (14 849 phrases) pour l'entraînement, 30 documents (863 phrases) pour le développement et 40 documents (672 phrases) pour le test. De même, nous considérons qu'une mention d'événement est correcte si son type d'événement, son sous-type et son empan correspondent à ceux d'une mention de référence. Nous utilisons les micro-mesures de précision, rappel et F1-mesure (F1) pour évaluer la performance globale.

Paramètres des modèles Pour le CNN mot, la taille de la fenêtre de contexte est de 31 mots. Les filtres de convolution ont pour leur part une dimension de 1, 2 et 3 mots et 300 filtres sont utilisés pour chaque dimension. Après chaque couche convolutive, initialisée selon un schéma orthogonal (Saxe *et al.*, 2014), une couche non linéaire *ReLU* est appliquée. Nous employons un abandon (*dropout*) de probabilité 0,5 après la couche initiale des plongements et de probabilité 0,3 après la concaténation du résultat des convolutions. La dimensionnalité des plongements de positions est de 50, à l'instar de (Nguyen & Grishman, 2015). Enfin, nous avons utilisé les plongements de mots préentraînés construits avec Word2vec sur le corpus Google News (Mikolov *et al.*, 2013).

Pour le CNN caractère, l'entrée est constituée de séquences de 1 024 caractères. Nous considérons tous les caractères sauf l'espace. La taille des filtres de convolution va de 2 à 10, avec 300 filtres par taille. La non-linéarité et l'initialisation de la couche convolutive sont les mêmes que pour le CNN mot. Les plongements de caractères comportent 300 dimensions et sont initialisés sur la base d'une distribution normale. Un abandon de 0,5 est réalisé après les plongements de caractères. Lors de l'entraînement conjoint dans le modèle de fusion précoce, les vecteurs de traits obtenus après les convolutions des deux modèles sont concaténés et comme pour le CNN mot, un abandon de 0,3 est appliqué avant la couche softmax.

Résultats et discussion Nous comparons notre modèle avec plusieurs modèles neuronaux proposés pour la même tâche n'utilisant pas de ressources externes : des modèles convolutifs (Nguyen & Grishman, 2015; Chen *et al.*, 2015; Nguyen *et al.*, 2016b; Nguyen & Grishman, 2018), des modèles récurrents (Nguyen *et al.*, 2016a; Zhao *et al.*, 2018), des modèles hybrides (Feng *et al.*, 2016), le modèle GAIL-ELMo (Zhang *et al.*, 2019) et un modèle fondé sur un mécanisme d'attention multilingue (Liu *et al.*, 2018). Nous ne considérons pas pour des raisons de comparabilité les modèles utilisant des ressources externes tels que (Bronstein *et al.*, 2015; Li *et al.*, 2019) ou (Yang *et al.*, 2019). Nous nous comparons également aux modèles plus récents fondés sur BERT tels que le modèle de (Wadden *et al.*, 2019) conjuguant BERT et un LSTM pour capturer un contexte intra et inter-phrastique et définir de façon plus dynamique les mentions candidates, le modèle BERT-QA (Du & Cardie, 2020), qui aborde la détection d'événements comme une tâche de question-réponse et le modèle DMBERT (Wang *et al.*, 2019), qui s'appuie sur l'apprentissage adverse pour mettre en œuvre une approche faiblement supervisée. Nous comparons également notre modèle avec 4 approches de base reposant sur BERT, en abordant la détection d'événements de manière similaire à la reconnaissance d'entités nommées dans (Devlin *et al.*, 2019) et avec les mêmes valeurs d'hyperparamètres.

La meilleure performance (F1 = 75,8 %) est obtenue en combinant les plongements de mots et de positions avec les plongements de caractères selon une stratégie de fusion tardive. Le tableau 3 montre également que l'ajout de plongements de caractères dans une stratégie de fusion tardive est plus performant que tous les modèles s'appuyant sur les mots, y compris les architectures complexes

Approches	**Précision**	**Rappel**	**F1**
Word CNN (Nguyen & Grishman, 2015)	71,8	66,4	69,0
Dynamic multi-pooling CNN (Chen *et al.*, 2015)	75,6	63,6	69,1
Joint RNN (Nguyen *et al.*, 2016a)	66,0	73,0	69,3
CNN with document context (Duan *et al.*, 2017)†	77,2	64,9	70,5
Non-Consecutive CNN (Nguyen *et al.*, 2016b)	na	na	71,3
Attention-based (Liu *et al.*, 2017)+	78,0	66,3	71,7
GAIL-ELMo (Zhang *et al.*, 2019)	74,8	69,4	72,0
Gated Cross-Lingual Attention (Liu *et al.*, 2018)	78,9	66,9	72,4
Graph CNN (Nguyen & Grishman, 2018)	77,9	68,8	73,1
Hybrid NN (Feng *et al.*, 2016)	84,6	64,9	73,4
DEEB-RNN3 (Zhao *et al.*, 2018)	72,3	75,8	74,0
BERT-base-uncased + LSTM (Wadden *et al.*, 2019)	na	na	68,9
BERT-base-uncased (Wadden *et al.*, 2019)	na	na	69,7
BERT-base-uncased (Du & Cardie, 2020)	67,2	73,2	70,0
BERT-QA (Du & Cardie, 2020)	71,1	73,7	72,4
DMBERT (Wang *et al.*, 2019)	77,6	71,8	74,6
DMBERT+Boot (Wang *et al.*, 2019)	77,9	72,5	75,1
BERT-base-uncased	71,7	68,5	70,0
BERT-base-cased	71,3	72,0	71,7
BERT-large-uncased	72,1	72,9	72,5
BERT-large-cased	69,3	**77,2**	73,1
CNN mot (équivalent à Word CNN)	71,4	65,9	68,5
CNN caractère	71,7	41,2	52,3
CNN mot + caractère - fusion précoce	**88,6**	61,9	72,9
CNN mot + caractère - fusion tardive	87,2	67,1	**75,8**

TABLE 3 – Évaluation de nos modèles et comparaison avec l'état de l'art pour la détection d'événements sur le test d'ACE 2005. †au-delà de la phrase, +avec les arguments de référence

s'appuyant sur les convolutions de graphe et les modèles exploitant BERT. Parmi ceux-ci, il est intéressant de noter que les modèles intégrant la casse (*cased*) sont plus performants que les modèles *uncased*, ce qui confirme l'importance de l'information portée par le niveau des caractères pour cette tâche, peut-être parce que la capitalisation est liée à la reconnaissance des entités nommées, qui sont généralement considérées comme importantes pour la détection des mentions d'événements. La similitude de nos résultats pour *BERT-base-uncased* avec ceux de (Du & Cardie, 2020) et (Wadden *et al.*, 2019) pour le même BERT accrédite par ailleurs la solidité de ce constat.

Cependant, nous pouvons constater que les plongements de caractères ne sont pas suffisants en eux-mêmes : en utilisant uniquement le CNN caractère, nous obtenons ainsi le plus petit rappel de toutes les approches considérées. Néanmoins, sa précision (71,7) est comparativement très élevée, ce qui confère une bonne fiabilité aux mentions qu'il détecte. Dans le cas de la fusion précoce, nous constatons que la précision est la plus élevée de tous les modèles comparés. Nous supposons que dans l'approche conjointe, l'influence des représentations fondées sur les caractères dépasse celle des plongements de mots et de positions et que la combinaison reproduit le déséquilibre entre la

Type d'événements	Nouvelles mentions trouvées	Mentions d'entraînement
End-Position	*steps*	*step*
Extradite	*extradited*	*extradition*
Attack	*wiped*	*wipe*
Start-Org	*creating*	*create*
Attack	*smash*	*smashed*
End-Position	*retirement*	*retire*

TABLE 4 – Nouvelles mentions trouvées grâce au modèle CNN mot+caractère (fusion tardive)

précision et le rappel observé pour le CNN caractère, le rappel étant le plus faible de tous les modèles à l'exception du CNN caractère. La fusion tardive permet un contrôle plus informé de la combinaison et, en donnant la priorité au CNN caractère pour déterminer le type des mentions identifiées par le CNN mot, la méthode tire profit de sa grande précision, permettant une augmentation de la précision de 71,7 à 87,2 tout en ayant un rappel élevé, passant de 65,9 pour le CNN mot à 67,1.

Finalement, nous avons mené une analyse plus qualitative en examinant les mentions d'événements nouvellement détectées par le modèle à fusion tardive comparativement au modèle à fusion précoce. Nous avons observé que parmi les 37 mentions concernées, certaines sont effectivement des variantes dérivationnelles ou flexionnelles de mots présents dans les données d'entraînement, comme illustré par le tableau 4. Ce constat semble confirmer que le modèle fondé sur les caractères peut capturer certaines informations sémantiques associées aux caractéristiques morphologiques des mots et parvenir ainsi à détecter de nouvelles mentions d'événements en relation avec des mentions d'entraînement. La présence dans le CNN caractère de filtres convolutifs d'une taille entre 2 et 10, c'est-à-dire couvrant une plage assez large de n-grammes de caractères, contribue très certainement à cette capacité.

4 Conclusion et perspectives

Dans cet article, nous avons étudié l'intégration de plongements de caractères dans un modèle neuronal de détection d'événements fondé un simple modèle CNN en testant des stratégies de fusion précoce ou tardive. Les meilleurs résultats sont obtenus en combinant les représentations fondées sur les mots avec celles fondées sur les caractères dans une stratégie de fusion tardive donnant la priorité au modèle de caractères pour décider du type d'événements. Cette méthode est plus performante que des approches plus complexes fondées sur les convolutions de graphe, les réseaux antagonistes ou les modèles BERT. Ces résultats montrent aussi qu'un modèle de caractères permet de surmonter certains problèmes concernant les mots nouveaux ou mal orthographiés dans les données de test.

Ce travail ouvre la voie à des études plus larges sur le problème de la robustesse des modèles de détection d'événements vis-à-vis des variations touchant les déclencheurs événementiels. De ce point de vue, il serait intéressant de tester si des modèles de langue de type Transformer s'appuyant sur les caractères (El Boukkouri *et al.*, 2020; Ma *et al.*, 2020), ou même s'affranchissant de la segmentation en mots (Clark *et al.*, 2021), pourraient s'avérer plus robustes qu'un modèle de type BERT.

Remerciements Ce travail a été partiellement soutenu par le programme européen Horizon 2020 au travers des projet NewsEyes (770299) et Embeddia (825153).

Références

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.

BRONSTEIN O., DAGAN I., LI Q., JI H. & FRANK A. (2015). Seed-Based Event Trigger Labeling : How far can event descriptions get us ? In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, p. 372–376.

CHEN Y., XU L., LIU K., ZENG D. & ZHAO J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, p. 167–176.

CLARK J. H., GARRETTE D., TURC I. & WIETING J. (2021). Canine : Pre-training an efficient tokenization-free encoder for language representation. *arXiv preprint arXiv :1602.02410*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2019)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics.

DU X. & CARDIE C. (2020). Event Extraction by Answering (Almost) Natural Questions. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 671–683, Online.

DUAN S., HE R. & ZHAO W. (2017). Exploiting Document Level Information to Improve Event Detection via Recurrent Neural Networks. In *Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017)*, p. 352–361 : Asian Federation of Natural Language Processing.

EL BOUKKOURI H., FERRET O., LAVERGNE T., NOJI H., ZWEIGENBAUM P. & TSUJII J. (2020). CharacterBERT : Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *28^{th} International Conference on Computational Linguistics (COLING 2020)*, p. 6903–6915, Barcelona, Spain (Online : International Committee on Computational Linguistics.

FENG X., HUANG L., TANG D., JI H., QIN B. & LIU T. (2016). A language-independent neural network for event detection. In *54th Annual Meeting of the Association for Computational Linguistics*, p. 66–71.

JI H., GRISHMAN R. *et al.* (2008). Refining Event Extraction through Cross-Document Inference. In *46th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 254–262.

JOZEFOWICZ R., VINYALS O., SCHUSTER M., SHAZEER N. & WU Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv :1602.02410*.

KIM Y., JERNITE Y., SONTAG D. & RUSH A. M. (2016). Character-Aware Neural Language Models. In *Thirtieth AAAI Conference on Artificial Intelligence*, p. 2741–2749.

LI Q., JI H. & HUANG L. (2013). Joint Event Extraction via Structured Prediction with Global Features. In *51st Annual Meeting of the Association for Computational Linguistics*, p. 73–82.

LI W., CHENG D., HE L., WANG Y. & JIN X. (2019). Joint event extraction based on hierarchical event schemas from FrameNet. *IEEE Access*, **7**, 25001–25015.

LIAO S. & GRISHMAN R. (2010). Using document level cross-event inference to improve event extraction. In *48th Annual Meeting of the Association for Computational Linguistics*, p. 789–797 : Association for Computational Linguistics.

LIU J., CHEN Y., LIU K. & ZHAO J. (2018). Event Detection via Gated Multilingual Attention Mechanism. In *Thirty-second AAAI Conference on Artificial Intelligence (AAAI-18)*.

LIU S., CHEN Y., LIU K. & ZHAO J. (2017). Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms. In *55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, p. 1789–1798, Vancouver, Canada.

LUONG M.-T. & MANNING C. D. (2016). Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, p. 1054–1063, Berlin, Germany.

MA W., CUI Y., SI C., LIU T., WANG S. & HU G. (2020). CharBERT : Character-aware pre-trained language model. In *28th International Conference on Computational Linguistics (COLING 2020)*, p. 39–50, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.4.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR 2013), workshop track*.

NGUYEN T. H., CHO K. & GRISHMAN R. (2016a). Joint Event Extraction via Recurrent Neural Networks. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 300–309.

NGUYEN T. H., FU L., CHO K. & GRISHMAN R. (2016b). A two-stage approach for extending event detection to new types via neural networks. *1st Workshop on Representation Learning for NLP*, p. 158.

NGUYEN T. H. & GRISHMAN R. (2015). Event Detection and Domain Adaptation with Convolutional Neural Networks. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, p. 365–371.

NGUYEN T. H. & GRISHMAN R. (2018). Graph Convolutional Networks With Argument-Aware Pooling for Event Detection. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*.

PETERS M., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep Contextualized Word Representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2018)*, p. 2227–2237, New Orleans, Louisiana : Association for Computational Linguistics.

SAXE A. M., MCCLELLAND J. L. & GANGULI S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *2nd International Conference on Learning Representations (ICLR 2014*.

SUN L., HASHIMOTO K., YIN W., ASAI A., LI J., YU P. & XIONG C. (2020). Adv-BERT : BERT is not robust on misspellings ! Generating nature adversarial samples on BERT. *arXiv preprint arXiv :2003.04985*.

WADDEN D., WENNBERG U., LUAN Y. & HAJISHIRZI H. (2019). Entity, relation, and event extraction with contextualized span representations. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, p. 5784–5789, Hong Kong, China.

WANG X., HAN X., LIU Z., SUN M. & LI P. (2019). Adversarial training for weakly supervised event detection. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT 2019)*, p. 998–1008.

YANG S., FENG D., QIAO L., KAN Z. & LI D. (2019). Exploring Pre-trained Language Models for Event Extraction and Generation. In *57th Annual Meeting of the Association for Computational Linguistics*, p. 5284–5294.

ZHANG T., JI H. & SIL A. (2019). Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, **1**(2), 99–120.

ZHAO Y., JIN X., WANG Y. & CHENG X. (2018). Document embedding enhanced event detection with hierarchical and supervised attention. In *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, p. 414–419 : Association for Computational Linguistics.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Lemmatization of Historical Old Literary Finnish Texts in Modern Orthography

Mika Hämäläinen[1] Niko Partanen[2] Khalid Alnajjar[1]
(1) Department of Digital Humanities
(2) Department of Finnish, Finno-Ugrian and Scandinavian Studies
(1,2) University of Helsinki, Finland
firstname.lastname@helsinki.fi

RÉSUMÉ

Les textes écrits en vieux finnois littéraire représentent la première œuvre littéraire jamais écrite en finnois à partir du XVIe siècle. Il y a eu plusieurs projets en Finlande qui ont numérisé des anciennes collections de textes et qui les ont rendues disponibles pour la recherche. Cependant, l'utilisation de méthodes TAL modernes avec telles données pose de grands défis. Dans cet article, nous proposons une approche pour normaliser et lemmatiser simultanément des textes écrits en vieux finnois littéraire à l'orthographe moderne. Notre meilleur modèle donne une précision de 96,3 % avec les textes écrits par Agricola et de 87,7 % avec d'autres textes contemporains hors du domaine. Notre méthode est publiée gratuitement sur Zenodo et Github.

ABSTRACT

Texts written in Old Literary Finnish represent the first literary work ever written in Finnish starting from the 16th century. There have been several projects in Finland that have digitized old publications and made them available for research use. However, using modern NLP methods in such data poses great challenges. In this paper we propose an approach for simultaneously normalizing and lemmatizing Old Literary Finnish into modern spelling. Our best model reaches to 96.3% accuracy in texts written by Agricola and 87.7% accuracy in other contemporary out-of-domain text. Our method has been made freely available on Zenodo and Github.

MOTS-CLÉS : données historiques, normalisation, lemmatisation.

KEYWORDS: historical data, normalization, lemmatization.

1 Introduction

Finnish language has a long literary history starting from the 16th history. A large portion of the books printed in Finnish is currenlty openly available for research, and especially the bibliographic record they form has already been studied. (Lahti *et al.*, 2019) investigated the document size and language in the bibliographic metadata records, and (Tolonen *et al.*, 2019, 58) took into account a number of other metadata fields including the titles, and recognize the lack of full-text documents as a downside of investigation that is possible. Although the whole body of books printed in Finland is not available as full-text, there are numerous smaller corpora that can already be used in various ways.

Volume 1 : conférence principale, pages 189–198.

Both the National Library of Finland and Institute for the Languages of Finland have produced a large number of digitized material from the era of the Old Literary Finnish. Former has made a large amount of scanned and text recognized publications available (National Library of Finland, 2021), and latter has created large plain text corpora from selected works (Institute for the Languages of Finland, 2014). The Agricola corpus used in this study has been morpho-syntactically annotated, and is to our knowledge the only annotated resource in the Old Literary Finnish (Institute for the Languages of Finland & University of Turku, 2020). Another very central resources is the Dictionary of Old Literary Finnish. [1] Thus far the Old Literary Finnish has largely eluded any attempts of NLP research as the historical written form cannot be processed easily with currently available NLP tools for Finnish, as they are designed for modern Finnish. We present our approach for normalizing and lemmatizing Old Literary Finnish automatically to modern Finnish orthography. As the resources for processing historical Finnish text are scarce, we have released the models presented in this paper on Zenodo [2] and through an easy-to-use Python library [3].

The use of existing NLP tools targeted for modern Finnish on historical materials can be made possible through normalization. Previous work conducted on English data indicates that normalization is a viable way of improving the accuracy of NLP methods such as POS tagging (van der Goot *et al.*, 2017). Another direction of digital humanities study has benefited from normalization of historical data in studying the use of neologisms in old letters (Säily *et al.*, 2018; Säily *et al.*, 2021). In their approach, without normalization, they would have been able to cover only a small subset of the corpus. The same corpus has also been studied without NLP tools (Nevalainen, 2021).

The history of printed written Finnish starts from the 16th century with the works of Mikael Agricola. His primer and religious works were followed by a continuous increase in the amount of the written Finnish materials. The language form used by Agricola is known as Old Literary Finnish (*vanha kirjasuomi*). The majority of the early Finnish publications included religious materials, although the text types started to diversify already in the 18th century. The majority of the oldest texts are translations. The period of Old Literary Finnish is often estimated to have lasted until 1810, after which the written Finnish started to transform into Early Modern Finnish. Eventual changes in printing laws and printing technology, which expanded the amount and variation of the printed materials, and the creation of regular Finnish newspapers, contributed to the stabilization of the written standard.

Linguistically one of the exceptional features of Old Literary Finnish is the variation it displays. The orthography was not yet entirely established, and there was extensive spelling variation. The age of these materials adds also a historical dimension, as there are linguistic features that are not present in the modern Finnish, or exist currently only in the dialects.

Our main contributions in this work are :

— Building the first artificial neural network model for normalizing historical Finnish.
— Conducting an evaluation for assessing the performance of the model on historical data from 1) the same source of the data used in building the model and 2) external out-of-domain historical data. In both cases, a high accuracy is achieved by the model.
— Publishing a user-friendly Python library that permits an instant usability of our model.

The target language form in our work is modern Finnish. Our model primarily lemmatizes, but since the output is harmonized into contemporary orthographic forms, the work is closely connected to the normalization task as well, and the output can be considered as one type of a normalization. The exact

1. https ://kaino.kotus.fi/vks/
2. https ://zenodo.org/record/4734143
3. https ://github.com/mikahama/murre

lemmatization choices and conventions were decided at the level of the original morpho-syntactic database, and we followed those closely also when our own additional test material was created. In later research also the morpho-syntactic annotations present in the corpus could be taken into account to further enrich the analysis. However, at the current stage a successful lemmatization is already a large improvement in available NLP methods.

This paper is structured as follows. We begin by describing the related work. Thereafter, the details of the data used to build the neural model are given, followed by the architecture and hyperparameters of the neural model. Section 5 presents the results and evaluation where we explain the different training strategies we experimented with and their performance against a baseline (historical Omorfi). Lastly, we discuss and conclude our work while highlighting future directions with a potentially great impact on humanities research such as automatic analysis of historical Finnish.

2 Related Work

Historical text normalization has been studied in the past for other languages than Finnish. A recent literature review (Bollmann, 2019) finds that there are five categories in which modern normalization approaches can be divided : substitution lists like VARD (Rayson *et al.*, 2005) and Norma (Bollmann, 2012), rule-based methods (Baron & Rayson, 2008; Porta *et al.*, 2013), edit distance based approaches (Hauser & Schulz, 2007; Amoia & Martinez, 2013), statistical methods and most recently neural methods (Partanen *et al.*, 2019; Duong *et al.*, 2020).

Statistical machine translation (SMT) based methods have been the most successful ones in the past in terms of statistical methods. The key idea behind these methods is to approach the task as a character-level machine translation problem, where a word is translated character by character to its normalized form. These methods have been applied to historical text (Pettersson *et al.*, 2013; Hämäläinen *et al.*, 2018) and dialect normalization (Samardzic *et al.*, 2015).

In the recent years, normalization has been approached as a character-level neural machine translation (NMT) problem similarly to the previous SMT approaches. The additional advantage is that a neural model does not need a separate language model like SMT does. Bollmann & Søgaard (2016) have shown that a bi-directional long short-term memory (bi-LSTM) can be used to normalize historical German texts. The paper presents a so-called multi-task learning setting where auxiliary data is added to improve the performance of the model. Multi-task learning setting generally improved the results. Their system outperformed the existing conditional random fields and Norma based approaches in terms of accuracy.

Text written in Uyghur language has been normalized with an LSTM and a noisy channel model (NCM) (Tursun & Cakici, 2017). They use a relatively small set of gold annotated data for training (around 200 hand normalized social media sentences). They augment this data by synthetically generating non-normalized text by introducing random changes in normalized text. In the same fashion, another research has used an LSTM model to normalize code-mixed data (Mandal & Nanmaran, 2018).

Recently Hämäläinen *et al.* (2019) have shown that bi-directional recurrent neural networks (BRNN) outperform regular unidirectional recurrent neural networks (RNN) when normalizing historical English data. Interestingly, additional layers and different attention models do not improve the results. Additional data such as time period, social metadata or pronunciation information in IPA characters

makes the results worse. According to them, post-processing can boost the accuracy of a character level NMT model more than changing the network structure. A simple dictionary filtering method improved the results.

Omorfi (Pirinen, 2015) is a popular rule-based tool used to do morphological analysis and lemmatization of modern Finnish. While Omorfi itself is not relevant for our work, there is a GitHub fork of the project known as Historical Omorfi [4]. The fork introduces several improvements to better cater for historical Finnish text. Currently, this tool is the only tool available for lemmatizing historical Finnish. Thereby we compare the results of our model to this tool.

3 Dataset of Historical Finnish

In order to use machine learning methods, data is needed to train a model. The data needs to have text written in Old Literary Finnish and its normalized lemmas. The lemmas should be aligned on a word level with the historical data in order to train the normalization more accurately. Fortunately, such a dataset exists. The corpus we use is *The Morpho-Syntactic Database of Mikael Agricola's Works* (Institute for the Languages of Finland & University of Turku, 2020) that contains 522,237 tokens and 38,222 sentences. The corpus includes all nine Finnish books translated by Mikael Agricola. The corpus is openly available in the Language Bank of Finland. [5] In our testing we also use the Dictionary of Old Literary Finnish [6] (Institute for the Languages of Finland, 2014), from which a small number of sentences has been sampled and manually normalized and lemmatized. Both resources are available under Creative Commons licenses. Since the *The Morpho-Syntactic Database of Mikael Agricola's Works* is licensed under CC BY-ND 4.0 (CLARIN PUB), we do not redistribute the training material ourselves, but it can be accessed in the Language Bank of Finland's concordance service [7]. Naturally, since this data is so old, it is already in Public Domain, and many of the original works are entirely openly accessible. For example, the Agricola's prayer book is available as high quality scans by the National Library of Finland [8].

Although there has not been prior use of this dataset in the computational linguistics, the corpus has been used in the linguistic studies. To illustrate this with few recent examples, Toropainen (2018) investigated the compounds in this variety, and Toropainen (2015) studied nouns that contain an initial adjective. Salmi (2020) discussed recently the German influence in the Agricola's language. Also annotating the corpus into it's current stage has been a long undertaking, and Inaba (2015) investigated the use of two Finnish cases with a prototype of the current database. It is beyond doubt that the materials of Old Literary Finnish still have much to contribute to the linguistic research. We believe that by creating new tools for natural language processing of these and similar materials we can further expand toward these goals. The research concerning Agricola's language and Old Literary Finnish in general is naturally much wider and has a long research history, especially in Finland, and we primarily wanted to illustrate in this section some of the previous studies where the same database was used.

4. https://github.com/jiemakel/omorfi/
5. urn:nbn:fi:lb-2019121804
6. https://kaino.kotus.fi/vks
7. https://korp.csc.fi
8. https://www.doria.fi/handle/10024/43445

4 Neural Normalization

In this section, we describe our artificial neural network model for normalizing and lemmatizing historical Finnish into standard Finnish. We begin by explaining how the dataset is used and how it is preprocessed. Thereafter, we elucidate the architecture of the model along with the technical details of its hyperparameters.

The corpus contains nine distinct works. In order to test the model's accuracy realistically, we selected seven books into the training data, and left the remaining two into the test set. Thereby the training data was 393,779 tokens and the test set 128,294 tokens. The books in the test set were specifically 'Messu eli Herran echtolinen' from 1549 and 'Rucouskiria' from 1544. Additionally 15% of the training data was used in the validation set. We considered it important not to select the test set from the entire corpus, as this would not give a clear picture about how much the model generalizes into new use cases, which would be the other books written in the same language variety. With the current sparsity of manually annotated data, the Agricola works in the currently used corpus, but which we kept unseen for the model, were the best option we had. We also extended the testing into other examples of Old Literary Finnish, but in a smaller scale. The data has the original written form of each sentence and a token level normalized lemma for each word. We use this parallel data to train our models.

We model the problem as a character level NMT problem. In practice, we split words into characters separated by white space and mark actual spaces between words with an under score (_). This allows to pass word boundary information to the model while the characters themselves are separated by spaces. We train the model to predict from the Old Literary Finnish word forms to the lemmas. As previous research (Partanen *et al.*, 2019) has found that using chunks of words instead of full sentences at a time improves the results, we train different models with different chunk sizes. This allows direct comparison of different chunk sizes. This way we train the models to predict one word at a time, two words at a time all the way to five words at a time. An example of the data can be seen in Table 1. This example is within the sentence A-II-369 in the corpus, which is located in the New Testament translation.

Size	**Source**	**Target**
Chunk 1	s y d h e m e n	s y d ä n
Chunk 3	p a l u e l l e n _ h e r r a _ c a i k e n	p a l v e l l a _ h e r r a _ k a i k k i

TABLE 1 – An example of the training data for chunk sizes 1 and 3.

We train all models using a bi-directional long short-term memory (LSTM) based model (Hochreiter & Schmidhuber, 1997) by using OpenNMT-py (Klein *et al.*, 2017) with the default settings except for the encoder where we use a BRNN (bi-directional recurrent neural network) (Schuster & Paliwal, 1997) instead of the default RNN (recurrent neural network), since BRNN based models have been shown to provide better results in a variety of tasks. We use the default of two layers for both the encoder and the decoder and the default attention model, which is the general global attention presented by Luong *et al.* 2015. The models are trained for the default of 100,000 steps. All models are trained with the same random seed (3435) to ensure reproducibility and to make their intercomparison possible.

5 Results and Evaluation

Our initial evaluation results were very good as seen in Table 2 where the accuracies are reported on a word level. The best model was the chunk of 3. The quality we reach is very high and on par with other comparable normalization tasks, such as in (Partanen *et al.*, 2019), which also corroborate the best performance at the chunk of three tokens. However, we are also interested in seeing how well our model works with other Old Literary Finnish texts. At any rate, we can see that our models outperform Historical Omorfi, which is the only tool publicly available for historical Finnish. As Omorfi produces all the possible lemmas for a given word, we count the accuracy based on if the correct lemma is in the list of the lemmas Omorfi produced for each word.

	Chunk 1	**Chunk 2**	**Chunk 3**	**Chunk 4**	**Chunk 5**	**Omorfi**
Accuracy	96.1%	96.2%	96.3%	96.2%	95.9%	40.5%

TABLE 2 – Token level accuracy of each model in the test data.

Because there is no other dataset freely available that would both be written in Old Literary Finnish and lemmatized to modern orthography, we take randomly 50 sentences from the example sentences of the dictionary of Old Literary Finnish [9] that is available online. Altogether these sentences have 562 words (excluding punctuation). We lemmatize these sentences with the model that has been trained with chunks of 3 as it worked the best out of the models and verify the lemmatization by hand. The results of this experiment are seen in Table 3.

	Chunk 3	**Omorfi**
Accuracy	87.7%	47.9%

TABLE 3 – Accuracy in the Old Literary Finnish dictionary sample.

The results drop in this evaluation, but it is only to be expected given that out-of-domain performance is typically lower for neural models. Nevertheless, we see clearly that the model does well in out-of-domain data and beats the current state of the art. The fact that Historical Omorfi gets better results in this dataset is a good indication that the text is very different from what is in the Agricola dataset. However, it clearly is not entirely distinct when we consider how well the model still performed.

If we look at the results more closely, analyzing the errors, we can see that the model usually does not predict non-words but rather words that are a part of the Finnish vocabulary. This indicates that the model has learned a good target representation. There are several errors in which the model has normalized the historical word correctly, but it has not lemmatized it, for example *runsast* was normalized to *runsaasti* although the lemma would be *runsas* 'plenty'. Another example is *ulosteon* that was already written as in the modern orthography was normalized unchanged to *ulosteon*, while the lemma is *ulosteko*. This example also illustrates how there is variation and historical change in spelling, as we find in the Agricola corpus comparable words spelled with the initial *v*, whereas in the later materials we find the variant above that is spelled closely the current standard. Analysing how this relates to the changes in characters used in different centuries is beyond the scope of our study, but it illustrates well the kind of variation we can find in the historical texts and their digitized versions.

9. https ://kaino.kotus.fi/vks

To analyze the errors further, the most typical source of problems are verbs that get lemmatized into nouns that look similar and vice versa. Thereby *olan* was lemmatized as *olla* 'to be' while it should have been *olka* 'shoulder'. Also, *nuole* was lemmatized as *nuoli* 'arrow', while the correct lemma is *nuolla* 'to lick'. Normalization to a wrong lemma within the same part-of-speech is also possible e.g. *kaipanne* to *kaivaa* 'to dig' instead of *kaivata* 'to miss'. Improving the recognition of such instances is a very important task for the future work, but the current accuracy also appears to be useful and satisfactory for many tasks, and is without doubt an improvement to the existing methods.

6 Conclusions and Future Work

Our results have a clear indication, both with in-domain and out-of-domain test data, of working successfully in lemmatizing Old Literary Finnish in the modern orthography. The models have been released on Zenodo and in a Python library [10]. By sharing the models we are making NLP research on historical Finnish data more widely accessible for the research community, as the currently available Historical Omorfi does not work well for texts that are this old. Our study also creates a benchmark into which the further work can easily be compared.

Having lower accuracy in another dataset is a reminder of the importance of evaluating normalization models on data that comes from a different distribution. This is something seldom seen in the previous work on historical spelling normalization. Despite this, the accuracy has remained relatively high and we have identified several possibly problematic phenomena in the test data that are more prone to errors. This error analysis helps in understanding the biases that using our model might introduce in historical data when it is used to lemmatize a corpus completely new to the model. Further research is needed to evaluate how the error rate varies when the distance grows to the materials of Agricola. It is beyond doubt that more diverse training material is needed to successfully process the entire corpus of Old Literary Finnish, but our study certainly has improved the position to initiate and continue such work.

The work we presented here also makes it possible for the current research on analyzing historical Finnish newspapers, such as (Jean-Caurant & Doucet, 2020; Kettunen *et al.*, 2020), to standardize post-OCR historical Finnish, which in turn permits employing state-of-the-art Finnish NLP methods and tools on such data (e.g. sentiment and semantic analysis (Hämäläinen & Alnajjar, 2019)).

When we work with the currently available resources, we must also remain aware that the digital versions have been edited in various ways, and do not contain necessarily all features of the original printed text (Toropainen, 2016, 175). Now when we increasingly have access also to the original prints as high quality scans, it is important to think how these different resources can be connected to one another. This will need a combination of both text recognition and NLP tools.

In the future, we are interested in conducting work on semantic change on historical data. This should be greatly facilitated by the fact that we can now considerably reliably lemmatize historical text. This means that training word embeddings models will become more accurate as the model is trained on lemmas instead of inflectional forms.

10. https ://github.com/mikahama/murre

Références

AMOIA M. & MARTINEZ J. M. (2013). Using comparable collections of historical texts for building a diachronic dictionary for spelling normalization. In *Proceedings of the 7th workshop on language technology for cultural heritage, social sciences, and humanities*, p. 84–89.

BARON A. & RAYSON P. (2008). VARD2 : A tool for dealing with spelling variation in historical corpora. In *Postgraduate conference in corpus linguistics*.

BOLLMANN M. (2012). (Semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, Lisbon, Portugal.

BOLLMANN M. (2019). A large-scale comparison of historical text normalization systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 3885–3898, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1389.

BOLLMANN M. & SØGAARD A. (2016). Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, p. 131–139, Osaka, Japan : The COLING 2016 Organizing Committee.

DUONG Q., HÄMÄLÄINEN M. & HENGCHEN S. (2020). An unsupervised method for OCR post-correction and spelling normalisation for Finnish. *arXiv preprint arXiv :2011.03502*.

HÄMÄLÄINEN M. & ALNAJJAR K. (2019). Let's FACE it. Finnish poetry generation with aesthetics and framing. In *Proceedings of the 12th International Conference on Natural Language Generation*, p. 290–300, Tokyo, Japan : Association for Computational Linguistics. DOI : 10.18653/v1/W19-8637.

HÄMÄLÄINEN M., SÄILY T., RUETER J., TIEDEMANN J. & MÄKELÄ E. (2018). Normalizing early English letters to present-day English spelling. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, p. 87–96.

HÄMÄLÄINEN M., SÄILY T., RUETER J., TIEDEMANN J. & MÄKELÄ E. (2019). Revisiting NMT for normalization of early English letters. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, p. 71–75, Minneapolis, USA : Association for Computational Linguistics. DOI : 10.18653/v1/W19-2509.

HAUSER A. W. & SCHULZ K. U. (2007). Unsupervised learning of edit distance weights for retrieving historical spelling variations. In *Proceedings of the First Workshop on Finite-State Techniques and Approximate Search*, p. 1–6.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

INABA N. (2015). *Suomen datiivigenetiivin juuret vertailevan menetelmän valossa*, volume 272 de *Mémoires de la Société Finno-Ougrienne*. Société Finno-Ougrienne.

INSTITUTE FOR THE LANGUAGES OF FINLAND (2014). Corpus of Old Literary Finnish. http ://urn.fi/urn :nbn :fi :lb-201407165.

INSTITUTE FOR THE LANGUAGES OF FINLAND & UNIVERSITY OF TURKU (2020). The Morpho-Syntactic Database of Mikael Agricola's Works version 1.1, Korp.

JEAN-CAURANT A. & DOUCET A. (2020). Accessing and investigating large collections of historical newspapers with the NewsEye platform. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, p. 531–532, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3383583.3398627.

KETTUNEN K., KOISTINEN M. & KERVINEN J. (2020). Ground truth OCR sample data of Finnish historical newspapers and journals in data improvement validation of a re-OCRing process. *Liber quarterly*. DOI : 10.18352/lq.10322.

KLEIN G., KIM Y., DENG Y., SENELLART J. & RUSH A. M. (2017). OpenNMT : Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL*. DOI : 10.18653/v1/P17-4012.

LAHTI L., MARJANEN J., ROIVAINEN H. & TOLONEN M. (2019). Bibliographic data science and the history of the book (c. 1500–1800). *Cataloging & Classification Quarterly*, **57**(1), 5–23.

LUONG M.-T., PHAM H. & MANNING C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv :1508.04025*.

MANDAL S. & NANMARAN K. (2018). Normalization of transliterated words in code-mixed data using Seq2Seq model & Levenshtein distance. In *Proceedings of the 2018 EMNLP Workshop W-NUT : The 4th Workshop on Noisy User-generated Text*, p. 49–53, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-6107.

NATIONAL LIBRARY OF FINLAND (2021). Digital Collections. https ://digi.kansalliskirjasto.fi.

NEVALAINEN T. (2021). Time's arrow reversed ? the (a)symmetry of language change. In M. HÄMÄLÄINEN, N. PARTANEN & K. ALNAJJAR, Éds., *Multilingual Facilitation*. Rootroo Ltd.

PARTANEN N., HÄMÄLÄINEN M. & ALNAJJAR K. (2019). Dialect text normalization to normative standard Finnish. In W. XU, A. RITTER, T. BALDWIN & A. RAHIMI, Éds., *The Fifth Workshop on Noisy User-generated Text (W-NUT 2019)*, p. 141–146, United States : The Association for Computational Linguistics.

PETTERSSON E., MEGYESI B. & TIEDEMANN J. (2013). An SMT approach to automatic annotation of historical text. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013 ; May 22-24 ; 2013 ; Oslo ; Norway. NEALT Proceedings Series 18*, volume 087, p. 54–69 : Linköping University Electronic Press.

PIRINEN T. A. (2015). Development and use of computational morphology of Finnish in the open source and open science era : Notes on experiences with Omorfi development. *SKY Journal of Linguistics*, **28**, 381–393.

PORTA J., SANCHO J.-L. & GÓMEZ J. (2013). Edit transducers for spelling variation in Old Spanish. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013 ; May 22-24 ; 2013 ; Oslo ; Norway. NEALT Proceedings Series 18*, volume 087, p. 70–79 : Linköping University Electronic Press.

RAYSON P., ARCHER D. & SMITH N. (2005). VARD versus WORD : a comparison of the UCREL variant detector and modern spellcheckers on english historical corpora. *Corpus Linguistics 2005*.

SÄILY T., MÄKELÄ E. & HÄMÄLÄINEN M. (2018). Explorations into the social contexts of neologism use in early English correspondence. *Pragmatics & Cognition*, **25**(1), 30–49. DOI : 10.1075/pc.18001.sai.

SALMI H. (2020). German influence on the Finnish in Mikael Agricola. *Finnish-German Yearbook of Political Economy, Volume 2*, p. 135.

SAMARDZIC T., SCHERRER Y. & GLASER E. (2015). Normalising orthographic and dialectal variants for the automatic processing of Swiss German. In *Proceedings of the 7th Language and Technology Conference*. ID : unige :82397.

SCHUSTER M. & PALIWAL K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, **45**(11), 2673–2681.

SÄILY T., MÄKELÄ E. & HÄMÄLÄINEN M. (2021). From plenipotentiary to puddingless : Users and uses of new words in early english letters. In M. HÄMÄLÄINEN, N. PARTANEN & K. ALNAJJAR, Éds., *Multilingual Facilitation*. Rootroo Ltd.

TOLONEN M., LAHTI L., ROIVAINEN H. & MARJANEN J. (2019). A quantitative approach to book-printing in Sweden and Finland, 1640–1828. *Historical methods : a journal of quantitative and interdisciplinary history*, **52**(1), 57–78.

TOROPAINEN T. (2015). Adjektiivialkuiset yhdyssubstantiivit Mikael Agricolan teoksissa. *Sananjalka*, **57**(1), 54–85.

TOROPAINEN T. (2016). Typografian vaikutus yhdyssubstantiivien oikeinkirjoitukseen agricolan teoksissa. *Sananjalka*, **58**(1), 175–198.

TOROPAINEN T. (2018). *Yhdyssanat ja yhdyssanamaiset rakenteet Mikael Agricolan teoksissa*. Thèse de doctorat, University of Turku. Turun yliopiston julkaisuja. Sarja C, Scripta lingua Fennica edita.

TURSUN O. & CAKICI R. (2017). Noisy Uyghur text normalization. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, p. 85–93, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/W17-4412.

VAN DER GOOT R., PLANK B. & NISSIM M. (2017). To normalize, or not to normalize : The impact of normalization on Part-of-Speech tagging. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, p. 31–39.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Méta-apprentissage : classification de messages en catégories émotionnelles inconnues en entraînement

Gaël Guibon[1, 2] Matthieu Labeau[1]
Hélène Flamein[2] Luce Lefeuvre[2] Chloé Clavel[1]
(1) LTCI, Télécom-Paris, Institut Polytechnique de Paris
(2) Direction Innovation & Recherche SNCF
prénom.nom@telecom-paris.fr, prénom.nom@sncf.fr

RÉSUMÉ

Dans cet article nous reproduisons un scénario d'apprentissage selon lequel les données cibles ne sont pas accessibles et seules des données connexes le sont. Nous utilisons une approche par méta-apprentissage afin de déterminer si les méta-informations apprises à partir de messages issus de médias sociaux, finement annotés en émotions, peuvent produire de bonnes performances une fois utilisées sur des messages issus de conversations, étiquetés en émotions avec une granularité différente. Nous mettons à profit l'apprentissage sur quelques exemples (*few-shot learning*) pour la mise en place de ce scénario. Cette approche se montre efficace pour capturer les méta-informations d'un jeu d'étiquettes émotionnelles pour prédire des étiquettes jusqu'alors inconnues au modèle. Bien que le fait de varier le type de données engendre une baisse de performance, notre approche par méta-apprentissage atteint des résultats décents comparés au référentiel d'apprentissage supervisé.

ABSTRACT

Meta-learning : Classifying Messages into Unseen Emotional Categories

In this paper, we place ourselves in a classification scenario in which the target data set classes and data type are not accessible during training. We use a meta-learning approach to determine whether or not meta-trained information from common social network data with fine-grained emotion labels can achieve competitive performance on conversation utterances labeled with different, higher level, emotions. We leverage few-shot learning to concur with the classification scenario. This approach proves to be effective for capturing meta-information from a source emotional tag set to predict previously unseen emotional tags. Even though shifting the data type triggers an expected performance drop, our meta-learning approach achieves decent results when compared to the fully supervised one.

MOTS-CLÉS : classification en émotions, méta-apprentissage, apprentissage sur peu d'exemples.

KEYWORDS: emotion classification, meta-learning, few shot learning, natural language processing.

1 Introduction

Le Traitement Automatique du Langage (TAL) fait souvent face à la nécessité de devoir entraîner un modèle de classification pour une tâche sans pour autant en posséder les annotations dédiées. C'est davantage le cas pour les entreprises dont les données spécialisées, souvent privées ou confidentielles, sont synonymes d'un processus d'annotation fastidieux et coûteux. Les données annotées en émotion entrent bien souvent dans ce cas de figure puisqu'elles sont majoritairement produites dans des

Volume 1 : conférence principale, pages 199–208.

conversations privées et restent délicates à annoter en raison de leur subjectivité. Nos travaux se concentrent sur la tâche de classification d'émotions de textes courts et informels pour lesquels nous faisons l'hypothèse qu'un méta-apprentissage puisse servir à la classification de textes qui divergent en structure langagière et en jeu d'étiquettes.

Classifier du texte en émotions fait l'objet de nombreux travaux allant de la classification en polarités (Strapparava & Mihalcea, 2007; Thelwall *et al.*, 2012; Yadollahi *et al.*, 2017) à l'usage de représentations émotionnelles plus précises et complexes (Alm *et al.*, 2005; Bollen *et al.*, 2009; Yu *et al.*, 2015; Zhang *et al.*, 2018a; Zhu *et al.*, 2019; Zhong *et al.*, 2019; Park *et al.*, 2019). Ces approches requièrent habituellement l'usage du maximum de données possible pour entraîner le modèle de classification. Toutefois, l'obtention de jeux de données conséquents n'est pas toujours possible pour une tâche dédiée. Le recours à des stratégies telles que l'apprentissage à partir de quelques exemples (*Few Shot Learning* – FSL) (Lake, 2015; Vinyals *et al.*, 2016; Ravi & Larochelle, 2016) ou le méta-apprentissage (Schmidhuber, 1987) semble alors nécessaire. Le méta-apprentissage consiste à "apprendre à apprendre", notamment en extrayant des méta-informations permettant une application plus efficace à de nouvelles tâches.

Ces deux approches ont émergé de la vision artificielle et y ont fait l'objet de différentes stratégies d'optimisation telles que l'apprentissage épisodique (Ravi & Larochelle, 2016), le méta-apprentissage indépendant du modèle (Finn *et al.*, 2017), ou encore l'apprentissage de métrique. Cette méthode a notamment donné lieu à plusieurs modèles, comme les réseaux siamois (Koch *et al.*, 2015), les réseaux concordants (Vinyals *et al.*, 2016), et les réseaux prototypiques (Snell *et al.*, 2017). Parmi ces approches, certaines ont été adaptées à des tâches du TAL (Bao *et al.*, 2020; Gao *et al.*, 2019b), et plus particulièrement à la classification de textes. Le méta-apprentissage à l'aide du FSL a notamment été utilisé pour entraîner un modèle de classification en sentiments en variant les 23 sujets du jeu de données d'Amazon (ARSC) (Yu *et al.*, 2018; Geng *et al.*, 2019; Bao *et al.*, 2020; Bansal *et al.*, 2020). Afin d'appliquer le méta-apprentissage à la classification en émotions, de multiples approches ont été récemment abordées. Nous pouvons notamment citer une approche par apprentissage de distribution (Zhang *et al.*, 2018b) à l'aide de la décomposition de plongements de phrases, associée aux K-voisins (KNN) les plus proches (Zhao & Ma, 2019), ainsi qu'une étude de l'ambiguïté des émotions par méta-apprentissage à l'aide de BiLSTM avec attention (Fujioka *et al.*, 2019).

Dans cet article nous nous plaçons dans une situation où nous n'avons pas accès aux données cibles et par conséquent cherchons à méta-apprendre sur des données que nous savons plus ou moins proches. Nous combinons méta-apprentissage et FSL pour prédire les émotions dans des énoncés informels issus de conversations de médias sociaux, et cherchons à vérifier que cela peut nous permettre d'obtenir un modèle performant, bien qu'appris sur des données différentes. De récents travaux ont montré que le méta-apprentissage peut s'appliquer en variant les thèmes abordés des commentaires Amazon (Bao *et al.*, 2020) ou les différentes relations entre entités du corpus dédié Few-Rel (Han *et al.*, 2018; Gao *et al.*, 2019a). Dans nos travaux, nous exploitons le méta-apprentissage non seulement lorsque le jeu d'étiquettes émotionnelles diffère, mais également lorsque le format et la structure des données diffèrent. Nous contribuons en implémentant un méta-apprentissage distinguant les données à la fois par leurs jeux d'étiquettes et par leur format. Avec ce scénario, nous montrons que le méta-apprentissage combiné à l'apprentissage à partir de peu d'exemples (FSL) peut être efficace sans avoir observé ces deux variables pendant l'entraînement. Le code est publiquement disponible sur Github : `https://github.com/gguibon/metalearning-emotion-datasource`.

2 Données et étiquettes

Nous considérons deux jeux de données différents, en anglais, afin de nous conformer à la mise en place d'un métamodèle appris sur des données différentes de celles sur lesquelles il sera évalué. Nous vérifions ainsi la qualité des méta-informations apprises et la capacité de transfert du métamodèle.

GoEmotions (Demszky *et al.*, 2020) est le jeu de données sources servant à l'apprentissage de méta-informations à partir d'étiquettes et de formats différents. Il s'agit d'un corpus de 58 000 commentaires Reddit étiquetés avec 27 catégories d'émotions, catégories que nous séparons en trois jeux d'étiquettes (*EmoTagSets*) afin de permettre un méta-apprentissage par la suite.

DailyDialog (Li *et al.*, 2017) représente le jeu de données cibles à étiqueter à l'aide de notre métamodèle appris en amont. Ce corpus est constitué de 13 118 conversations de "tous les jours" sur différents thèmes. Pour nos travaux, nous n'utilisons que les messages individuels sans contexte conversationnel. Pour appliquer notre modèle nous utilisons uniquement les couples message/étiquette issus de l'échantillon de test officiel [1], soit un ensemble de 1 419 messages pour 6 émotions (*EmoTagSet3* ci-après). DailyDialog est également fourni avec une séparation en trois échantillons par *ratio* train/val/test qui assurent la répartition de toutes les étiquettes. Bien que ces échantillons par *ratio* ne permettent pas de méta-apprentissage, nous les utilisons pour de l'apprentissage supervisé standard à des fins de comparaison avec le méta-apprentissage, ainsi qu'à l'optimisation des hyperparamètres.

Jeux d'étiquettes. Nous considérons 3 jeux d'étiquettes émotionnelles distinctes, construites en prenant en compte leur nombre d'occurrences total pour avoir des tailles plus ou moins similaires, ainsi que leur présence ou non dans le corpus de test. Pour le méta-entraînement, nous utilisons le jeu nommé *EmoTagSet1* constitué des étiquettes suivantes : *admiration, approval, annoyance, amusement, love, confusion, realization, excitement, remorse, nervousness* et *pride*. Pour la validation, nous nommons *EmoTagSet2* le jeu d'étiquettes suivantes : *gratitude, curiosity, disapproval, optimism, disappointment, caring, desire, embarrassment, relief* et *grief*. Pour le test, aussi bien sur GoEmotions que sur DailyDialog, le jeu nommé *EmoTagSet3* est constitué de 6 émotions dites "basiques" : *joy, sadness, anger, fear, disgust* et *surprise*.

Ainsi, les 27 étiquettes sont séparées en 3 jeux distincts. Mis ensemble, les jeux d'entraînement et de validation représentent 21 émotions exclusives à GoEmotions, tandis que les 6 émotions du jeu de test sont présentes aussi bien dans GoEmotions que dans DailyDialog et représentent des émotions "basiques" ayant une granularité que l'on peut considérer comme moins fine. Ce dernier jeu d'étiquettes rend également la comparaison des résultats possible.

3 Méthode et protocole expérimental

GoEmotions (Demszky *et al.*, 2020) est utilisé pour l'entraînement tandis que DailyDialog (Li *et al.*, 2017) l'est pour l'évaluation, jouant ainsi le rôle de données privées non étiquetées. L'objectif est de transférer les méta-informations apprises à partir de commentaires Reddit vers des messages de conversations de tous les jours, les deux sources divergeant ainsi en structure et en vocabulaire.

Méta-apprentissage. Dans un premier temps, nous mettons en place un méta-apprentissage d'émo-

1. L'étude du contexte conversationnel dépasse le cadre de cet article. Nous nous concentrons uniquement sur les messages, les différences de structure et de jeu d'étiquettes émotionnelles pour du méta-apprentissage.

tions à partir des échantillons d'entraînement et de validation de GoEmotions. Cela nous permet d'apprendre des méta-informations que nous évaluons ensuite à l'aide de l'échantillon de test de DailyDialog, suivant le principe présenté en Figure 1. Notre but est de méta-entraîner un modèle de classification à partir de quelques exemples : plus précisément, en utilisant du FSL avec seulement 5 exemples par classe émotionnelle issue du jeu d'entraînement de GoEmotions. Nous distinguons alors 3 jeux d'étiquettes distinctes : un pour l'entraînement (*EmoTagSet1*), un pour la validation (*EmoTagSet2*) et un pour le test (*EmoTagSet3*). Les classes utilisées en test correspondent aux 6 émotions présentes dans DailyDialog afin de les exclure de l'apprentissage et de pouvoir comparer les résultats.
Pour concevoir notre méta-modèle, nous utilisons les réseaux prototypiques (Snell *et al.*, 2017) avec un apprentissage épisodique (Ravi & Larochelle, 2016) afin de combiner méta-apprentissage et FSL.

Concrètement, un épisode est défini par trois paramètres : le nombre de classes N_c (*ways*), le nombre d'exemples d'entraînement N_s (*shots*) pour chaque classe et le nombre d'éléments à étiqueter N_q (*queries*). Les éléments de l'épisode sont projetés dans un espace vectoriel, dans lequel sont calculés les prototypes de classes, à l'aide d'un encodeur f_ϕ. Lors de chaque épisode, nous appliquons l'apprentissage de métrique aux quelques exemples visibles S_k (*shots*) d'une classe k (*way*) pour lui attribuer un prototype $\mathbf{c}_k$ qui correspond à la moyenne des exemples $\mathbf{x}_i \in S_k$ une fois encodés par f_ϕ. Un prototype de classe est donc égal à : $\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$ où N_C représente le nombre de classes. La fonction d'encodage f_ϕ est ensuite également appliquée aux éléments à étiqueter, soit le jeu de requêtes Q_k. Nous minimisons ensuite la distance euclidienne d entre les vecteurs des prototypes de chaque classe et les vecteurs issus de l'encodage des éléments à étiqueter $d\left(f_\phi(\mathbf{x}), \mathbf{c}_k\right)$. L'encodeur est mis à jour à l'aide du calcul de l'erreur suivant : $\frac{1}{N_C N_Q}\left[d\left(f_\phi(\mathbf{x}), \mathbf{c}_k\right) + \log \sum_{k'} \exp\left(-d\left(f_\phi(\mathbf{x}), \mathbf{c}_{k'}\right)\right)\right]$ où N_Q représente le nombre d'éléments à étiqueter (communément nommé *Query Set*). Un épisode se résume donc à créer un prototype pour chaque classe à partir de peu d'exemples avant d'assigner une classe à un élément à étiqueter en fonction du prototype de classe qui lui est le plus proche. Les réseaux prototypiques créent ainsi, par le biais de l'encodeur, des vecteurs représentant chaque classe à partir de quelques exemples du jeu de support S (*Support Set*), correspondant au jeu d'exemples aléatoires dans lequel nous obtenons N_s exemples pour chaque classe N_c. Dans le cas du méta-apprentissage, ces vecteurs, ou prototypes, sont ici utilisés pour prédire des classes jamais vues dans le jeu d'entraînement. L'encodeur est ainsi optimisé pour chercher à obtenir une représentation de méta-informations inhérentes à chaque classe.

Nous varions les encodeurs en considérant alternativement : la moyenne des plongements lexicaux des éléments de la classe (AVG), un encodeur convolutionnel (CNN) (Kim, 2014) ou un encodeur *Transformer* (Vaswani *et al.*, 2017) (Transfo). La Figure 1 montre une vision globale de notre stratégie de méta-apprentissage, de l'apprentissage à l'évaluation. Nous définissons la composition d'un épisode par $N_c = 6$, $N_s = 5$ et $N_q = 30$, signifiant alors une tâche d'apprentissage en 5 exemples aléatoires, 6 classes et 30 cibles aléatoires (*5-shot 6-way 30-query*) dans laquelle N_c est contraint par le nombre de classes en test, le modèle étant *in fine* testé sur les 6 émotions du jeu de test de DailyDialog.

La composition des épisodes pour l'entraînement et pour la validation est identique. Pour le test, en revanche, nous utilisons 1 000 épisodes aléatoires pour lesquels le jeu d'exemples cibles (*query set*) est choisi aléatoirement à partir du jeu de test tout en suivant le principe des 5 exemples cibles par classe (c'est-à-dire 5 exemples à étiqueter pour chacune des 6 émotions). Afin d'assurer l'évaluation du méta-apprentissage, ces 6 classes sont disponibles uniquement lors du test.

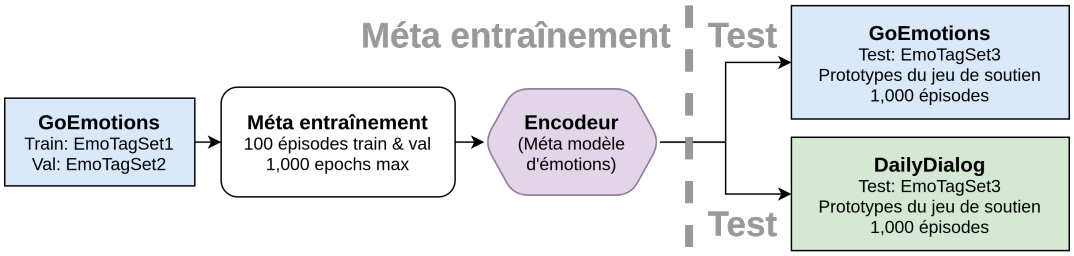


FIGURE 1 – Vue globale de la stratégie de méta-apprentissage. Lors du test sur DailyDialog, seuls les messages du jeu de test officiel sont pris en compte. EmoTagSet1 ∩ EmoTagSet2 ∩ EmoTagSet3 = ∅.

Notre protocole expérimental est le suivant : chaque jeu de données est pré-traité à l'aide du `TweetTokenizer` de NLTK [2] et de la normalisation en minuscules puis, pour les représentations textuelles de départ, nous utilisons les plongements lexicaux d'un modèle pré-entraîné de *fastText* (Joulin *et al.*, 2017), suivant ainsi des travaux antérieurs (Bao *et al.*, 2020). Alternativement, nous utilisons un modèle BERT pré-entraîné (Devlin *et al.*, 2019), bien que ce dernier n'améliore guère les résultats, confortant l'hypothèse de l'impact réduit sur des données de messages non séparés en phrases et qui possèdent une variation excessive du nombre de *tokens* (Bao *et al.*, 2020).

Apprentissage supervisé pour comparaison future. Dans un premier temps nous appliquons un apprentissage supervisé standard en utilisant uniquement les jeux de données officiels par *ratio* de DailyDialog (train/val/test). L'apprentissage supervisé sert donc de référence, démontrant les scores atteignables avec une approche classique et permettant ensuite une comparaison avec le méta-apprentissage. Dans cette approche supervisée, l'encodeur et le modèle de classification ne sont pas distincts puisqu'il s'agit simplement d'ajouter une couche cachée affine suivie d'un *softmax*, puis de calculer la log-vraisemblance négative pour obtenir une entropie croisée sur les différentes prédictions.

Hyper-paramètres. Pour rappel, nous répliquons ici un scénario dans lequel nous entraînons un modèle de classification sans avoir accès aux données cibles. Cependant, à des fins de comparaison, nous utilisons les hyper-paramètres obtenus à l'aide d'une recherche ciblée par quadrillage effectuée uniquement lors de l'apprentissage supervisé. Cela réduit la dépendance des résultats à d'éventuels paramètres spécifiques au méta-apprentissage et implique une évaluation plus fiable. Les hyper-paramètres généraux sont un pas d'apprentissage de $1e-3$, des vecteurs en entrée de taille 300, un abandon (*dropout*) de 10% sans coupe de gradient et un arrêt de l'apprentissage au bout de 20 *epochs* (*i.e.* un ensemble de 100 épisodes aléatoires). Les CNN suivent la configuration de Kim (Kim, 2014) avec des filtres ayant trois tailles différentes, de 3 à 5. Contrairement à Kim, nous utilisons ici 5 000 filtres au lieu de 50. Pour l'encodeur en *Transformer* le pas d'apprentissage est de $1e-4$, le *dropout* est augmenté à 20% et celui de l'encodeur positionnel est conservé à 10%.

Métriques d'évaluation. Nous évaluons les performances des modèles en nous inspirant de travaux précédents du FSL qui utilisent l'exactitude (Snell *et al.*, 2017; Sung *et al.*, 2018; Bao *et al.*, 2020). Toutefois, nous allons plus loin en considérant également une F1-mesure pondérée et le coefficient de corrélation de Matthews (MCC) (Cramir, 1946; Baldi *et al.*, 2000) comme suggéré par de récentes études en biologie (Chicco & Jurman, 2020), mais dans une version adaptée à une classification multi-classes (Gorodkin, 2004) qui convient davantage à notre tâche. Chaque résultat affiché représente la moyenne de l'ensemble des épisodes de test.

2. https://www.nltk.org/

4 Résultats

Apprentissage supervisé (hyperparamètres)
DailyDialog (6 classes)

Enc.	Clf.	Acc	F1	MCC
AVG	MLP	49,73	42,06	42,32
CNN	MLP	**62,57**	**54,89**	**59,12**
Transfo	MLP	55,35	48,52	49,24

Méta-apprentissage
GoEmotions : 6 way 5 shot 30 query

Enc.	Clf.	Acc ±	F1 ±	MCC ±
AVG	Proto	39,00 04, 8	38,35 04, 9	27,14 05, 8
CNN	Proto	42,83 04, 9	42,11 05, 0	31,76 05, 9
Dist.	Ridge	43,67 09, 8	42,71 09, 5	33,20 12, 0
Transfo	Proto	**95,89** 04, 3	**95,27** 05, 3	**95,31** 04, 8
Évaluation du métamodèle sur le jeu de test de DailyDialog : 1 000 épisodes				
AVG	Proto	19,60 03, 6	19,38 03, 7	03,40 04, 1
CNN	Proto	19,37 03, 6	18,77 03, 7	03,30 04, 3
Dist.	Ridge	26,05 08, 1	24,78 07, 9	11,58 10, 0
Transfo	Proto	**46,29** 25, 8	**39,02** 29, 6	**40,67** 30, 7

TABLE 1 – À gauche : apprentissage supervisé sur les messages de DailyDialog ; À droite : le méta-apprentissage entraîné en séparant les classes de GoEmotions, puis appliqué sur DailyDialog (test) ; évalués en exactitude (Acc), F1-mesure et coefficient de corrélation de Matthews (MCC). ± montre la variance au fil des épisodes.

Résultats de l'apprentissage supervisé. Les résultats présentés en partie gauche du tableau 1 proviennent de l'utilisation des jeux de données officiels de DailyDialog. Comme expliqué en Section 3, nous cherchons les meilleurs hyperparamètres pour chaque encodeur et chaque modèle de classification lors de cette phase d'apprentissage supervisé. Ceci s'avère nécessaire et sensible en particulier concernant les *transformers* (Vaswani *et al.*, 2017) qui requièrent un ajustement précis pour converger, d'autant plus quand, comme c'est le cas ici, le jeu de données est de taille relativement petite. La taille du jeu de données semble précisément être la raison pour laquelle le modèle de classification par *transformers* donne des résultats inférieurs à ceux du CNN dans cette approche purement supervisée.

Résultats du méta-apprentissage. La seconde section du tableau 1 présente deux résultats principaux : ceux de la phase de méta-apprentissage sur GoEmotions utilisant les jeux de données par ensemble d'étiquettes (11 émotions en entraînement, 10 autres en validation et 6 différentes en test) et l'évaluation de ces modèles sur le jeu de test officiel de DailyDialog (6 émotions). Comme on peut s'y attendre, le méta-apprentissage donne de moins bons résultats que l'apprentissage supervisé. Cela s'explique par l'apprentissage de méta-informations sur des sources variant en jeu d'étiquettes, en longueurs de phrases et ayant un contexte conversationnel différent. Les résultats obtenus montrent qu'une structure linguistique similaire entre le jeu de données cibles et celui d'entraînement facilite l'apprentissage de méta-informations, entraînant alors de meilleures performances. En effet, les résultats du méta-apprentissage obtenus sur GoEmotions sont meilleurs que ceux obtenus sur DailyDialog, bien qu'il s'agisse du même modèle. D'autre part, contrairement aux résultats de l'approche supervisée, l'utilisation d'un *Transformer* en tant qu'encodeur, ici associé aux réseaux prototypiques pour permettre un méta-entraînement, surpasse les autres encodeurs de manière significative. Cette approche surpasse également la baseline récente associant signatures de distribution (Dist.) sous forme d'attention à un Ridge Regressor (Ridge) (Bao *et al.*, 2020). Nous pensons que les piètres résultats obtenus en utilisant les CNN (Kim, 2014) en tant qu'encodeur démontrent le besoin d'attention dans le processus d'apprentissage de méta-informations pertinentes, d'autant plus avec une quantité réduite de données, ce qui confirmerait les précédents travaux faisant une observation similaire (Sun *et al.*, 2019).

5 Discussions

Méta-modèles et fonctionnement sur des émotions inconnues en entraînement. Les réseaux prototypiques utilisent le jeu de support (*support set*) pour calculer un prototype pour chaque classe (*way*), créant ainsi de nouveaux prototypes lors de chaque épisode. Cela signifie que l'encodeur entraîné ne dépend pas des classes prédites mais des informations regroupées déterminant la position des éléments dans l'espace vectoriel. La proximité permettant d'assigner un élément cible (*query*) à un prototype de classe est relative et, de ce fait, encoder un élément "loin" des prototypes dans l'espace vectoriel n'aura pas nécessairement d'impact sur la qualité de la prédiction.

Nature et ambiguïté des étiquettes émotionnelles. Il existe des liens non systématiques d'hyperonymie entre les émotions de base (*EmoTagSet3*) et les 21 émotions à granularité fine exclusives à GoEmotions et utilisées pour l'entraînement et la validation. Ces relations, fournies dans GoEmotions, peuvent être contestées. Ainsi, la joie est définie comme ayant notamment pour sous-catégories l'amusement et l'approbation ; tandis que la surprise inclut la curiosité. En outre, avec 95,27 % en F1-mesure, le méta-apprentissage fonctionne très bien sur GoEmotions (tableau 1) et semble ainsi prendre en compte l'ambiguïté et la différence de granularité des étiquettes.

Le méta-apprentissage à partir de différentes sources. Nous avons effectué un perfectionnement (*fine-tuning*) des modèles appris sur GoEmotions en utilisant le jeu de test de ce corpus (6 émotions), avant de l'appliquer sur DailyDialog. Ce perfectionnement était constitué d'une itération de 10 épisodes supplémentaires, contrairement aux 1 000 itérations maximales sur 100 épisodes utilisées lors de l'entraînement, et avait pour objectif de légèrement adapter l'encodeur au jeu d'étiquettes cibles en tirant parti des méta-informations apprises lors de l'entraînement. Le jeu d'étiquettes en test étant constitué des mêmes 6 émotions, ce perfectionnement nous permet principalement de vérifier si la difficulté de la tâche est due à la variété du jeu d'étiquettes ou à la variété des sources de données, qu'il s'agisse de GoEmotions (test) ou DailyDialog (test). Cette procédure a fourni des résultats de qualité moindre, nous conduisant à l'hypothèse selon laquelle les différences entre les structures langagières (messages de réseaux sociaux et messages informels journaliers) sont la principale source d'erreurs lors de l'utilisation de notre méta-apprentissage.

6 Conclusion

Nous avons mis en place un scénario de classification dans lequel nous ne possédons qu'un seul type de données d'entraînement, sans garantie de la similarité des données de test aussi bien au niveau du format que du jeu d'étiquettes. Nous avons utilisé des messages issus de médias sociaux étiquetés en émotions fines pour l'obtention de méta-informations à l'aide de la combinaison de méta-entraînement et d'entraînement à partir de peu d'exemples, évaluée sur des messages issus de conversations et comportant un jeu d'étiquettes différent. Pour cela nous avons mis en place des réseaux prototypiques avec pour encodeur un *Transformer*, appris par approche épisodique afin de simuler l'accès restreint aux données. Nous avons ainsi obtenu des résultats encourageants si l'on compare les performances entre le méta-modèle et le modèle de référence appris de manière supervisée. Cette approche fonctionne dans le cas de l'apprentissage de méta-informations liées aux différentes émotions mais peine à s'adapter à la variation de source de données. Par la suite, nous souhaitons vérifier dans de futurs travaux si le méta-entraînement peut être étendu à un cadre de classification utilisant le contexte conversationnel précédent un message.

Références

ALM C. O., ROTH D. & SPROAT R. (2005). Emotions from text : machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, p. 579–586, Vancouver, British Columbia, Canada : Association for Computational Linguistics. DOI : 10.3115/1220575.1220648.

BALDI P., BRUNAK S., CHAUVIN Y. & NIELSEN H. (2000). Assessing the accuracy of prediction algorithms for classification : An overview. *Bioinformatics*, **16**(5), 412–424. DOI : 10.1093/bioinformatics/16.5.412.

BANSAL T., JHA R., MUNKHDALAI T. & MCCALLUM A. (2020). Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 522–534, Online : Association for Computational Linguistics.

BAO Y., WU M., CHANG S. & BARZILAY R. (2020). Few-shot text classification with distributional signatures. In *International Conference on Learning Representations*.

BOLLEN J., PEPE A. & MAO H. (2009). Modeling public mood and emotion : Twitter sentiment and socio-economic phenomena. *arXiv :0911.1583 [cs]*. arXiv : 0911.1583.

CHICCO D. & JURMAN G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, **21**(1), 6.

CRAMIR H. (1946). Mathematical methods of statistics. *Princeton U. Press, Princeton*, **500**.

DEMSZKY D., MOVSHOVITZ-ATTIAS D., KO J., COWEN A., NEMADE G. & RAVI S. (2020). GoEmotions : A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4040–4054, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.372.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

FINN C., ABBEEL P. & LEVINE S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In D. PRECUP & Y. W. TEH, Éds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 de *Proceedings of Machine Learning Research*, p. 1126–1135, International Convention Centre, Sydney, Australia : PMLR.

FUJIOKA T., BERTERO D., HOMMA T. & NAGAMATSU K. (2019). Addressing ambiguity of emotion labels through meta-learning. *CoRR*, **abs/1911.02216**.

GAO T., HAN X., LIU Z. & SUN M. (2019a). Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 6407–6414. DOI : 10.1609/aaai.v33i01.33016407.

GAO T., HAN X., ZHU H., LIU Z., LI P., SUN M. & ZHOU J. (2019b). FewRel 2.0 : Towards More Challenging Few-Shot Relation Classification. *arXiv :1910.07124 [cs]*. arXiv : 1910.07124.

GENG R., LI B., LI Y., ZHU X., JIAN P. & SUN J. (2019). Induction Networks for Few-Shot Text Classification. *arXiv :1902.10482 [cs]*. arXiv : 1902.10482.

GORODKIN J. (2004). Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, **28**(5-6), 367–374.

HAN X., ZHU H., YU P., WANG Z., YAO Y., LIU Z. & SUN M. (2018). FewRel : A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. *arXiv :1810.10147 [cs, stat]*. arXiv : 1810.10147.

JOULIN A., GRAVE E., BOJANOWSKI P. & MIKOLOV T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, p. 427–431, Valencia, Spain : Association for Computational Linguistics.

KIM Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv :1408.5882*.

KOCH G., ZEMEL R. & SALAKHUTDINOV R. (2015). Siamese Neural Networks for One-shot Image Recognition. *ICML*, p.$\tilde{8}$.

LAKE B. (2015). LakeEtAl2015Science-startOfFewShot.pdf. *Sciences Mag*.

LI Y., SU H., SHEN X., LI W., CAO Z. & NIU S. (2017). Dailydialog : A manually labelled multi-turn dialogue dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

PARK S., KIM J., JEON J., PARK H. & OH A. (2019). Toward dimensional emotion detection from categorical emotion annotations. *arXiv preprint arXiv :1911.02499*.

RAVI S. & LAROCHELLE H. (2016). Optimization as a model for few-shot learning.

SCHMIDHUBER J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn : the meta-meta-... hook*. Thèse de doctorat, Technische Universität München.

SNELL J., SWERSKY K. & ZEMEL R. (2017). Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, p. 4077–4087.

STRAPPARAVA C. & MIHALCEA R. (2007). Semeval-2007 task 14 : Affective text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, p. 70–74.

SUN S., SUN Q., ZHOU K. & LV T. (2019). Hierarchical Attention Prototypical Networks for Few-Shot Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 476–485, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1045.

SUNG F., YANG Y., ZHANG L., XIANG T., TORR P. H. & HOSPEDALES T. M. (2018). Learning to compare : Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1199–1208.

THELWALL M., BUCKLEY K. & PALTOGLOU G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, **63**(1), 163–173.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

VINYALS O., BLUNDELL C., LILLICRAP T., WIERSTRA D. *et al.* (2016). Matching networks for one shot learning. In *Advances in neural information processing systems*, p. 3630–3638.

YADOLLAHI A., SHAHRAKI A. G. & ZAIANE O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, **50**(2), 1–33.

YU L.-C., WANG J., LAI K. R. & ZHANG X.-J. (2015). Predicting Valence-Arousal Ratings of Words Using a Weighted Graph Method. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, p. 788–793, Beijing, China : Association for Computational Linguistics. DOI : 10.3115/v1/P15-2129.

YU M., GUO X., YI J., CHANG S., POTDAR S., CHENG Y., TESAURO G., WANG H. & ZHOU B. (2018). Diverse Few-Shot Text Classification with Multiple Metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1206–1215, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1109.

ZHANG Y., FU J., SHE D., ZHANG Y., WANG S. & YANG J. (2018a). Text Emotion Distribution Learning via Multi-Task Convolutional Neural Network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, p. 4595–4601, Stockholm, Sweden : International Joint Conferences on Artificial Intelligence Organization. DOI : 10.24963/ijcai.2018/639.

ZHANG Y., FU J., SHE D., ZHANG Y., WANG S. & YANG J. (2018b). Text emotion distribution learning via multi-task convolutional neural network. In *IJCAI*, p. 4595–4601.

ZHAO Z. & MA X. (2019). Text emotion distribution learning from small sample : A meta-learning approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3957–3967, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1408.

ZHONG P., WANG D. & MIAO C. (2019). Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations. *arXiv :1909.10681 [cs]*. arXiv : 1909.10681.

ZHU S., LI S. & ZHOU G. (2019). Adversarial Attention Modeling for Multi-dimensional Emotion Regression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 471–480, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1045.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Prédire l'aspect linguistique en anglais au moyen de *transformers*

Eleni Metheniti[1, 2] Tim van de Cruys[3] Nabil Hathout [1]
(1) CLLE, CNRS & Université Toulouse - Jean Jaurès, France
(2) IRIT, CNRS & Université Toulouse - Paul Sabatier, France
(3) Faculté des Lettres, KU Leuven, Belgique

RÉSUMÉ

L'aspect du verbe décrit la manière dont une action, un événement ou un état exprimé par un verbe est lié au temps ; la télicité est la propriété d'un syntagme verbal qui présente une action ou un événement comme étant mené à son terme ; la durée distingue les verbes qui expriment une action (dynamique) ou un état (statique). Ces caractéristiques essentielles à l'interprétation du langage naturel, sont également difficiles à annoter et à identifier par les méthodes de TAL. Dans ce travail, nous estimons la capacité de différents modèles de type *transformers* pré-entraînés (BERT, RoBERTa, XLNet, ALBERT) à prédire la télicité et la durée. Nos résultats montrent que BERT est le plus performant sur les deux tâches, tandis que les modèles XLNet et ALBERT sont les plus faibles. Par ailleurs, les performances de la plupart des modèles sont améliorées lorsqu'on leur fournit en plus la position des verbes. Globalement, notre étude établit que les modèles de type *transformers* captent en grande partie la télicité et la durée.

ABSTRACT

Classifying Linguistic Aspect in English with Transformers

Verb aspect describes how an action, event, or state of a verb relates to time ; telicity focuses on whether the verb's action or state has an end point or not (telic/atelic), and duration denotes whether a verb expresses an action (dynamic) or a state (stative). These features are integral to the interpretation of natural language, but also hard to annotate and identify with NLP methods. In this work, we explore whether different kinds of fine-tuned transformer models (BERT, RoBERTa, XLNet, ALBERT) are successful in the task of binary classification of telicity and duration. Both for telicity and duration, BERT is the most successful, while certain XLNet and ALBERT models completely failed at the classification task. The use of verb position vectors significantly improves performance in most models. The results show that transformers models adequately capture telicity and duration.

MOTS-CLÉS : transformers, apprentissage automatique, aspect lexical, télicité, durée.

KEYWORDS: transformers, machine learning, lexical aspect, telicity, duration.

1 Introduction

L'aspect est une propriété temporelle des actions, des événements et des états décrits par les verbes, au-delà du temps verbal. Il englobe différentes propriétés, telles que la **télicité** et la **durée**. L'action du verbe est dite *télique* si elle a un point final. Lorsque le verbe désigne un état ou lorsque l'accomplissement de l'action du verbe est impossible, qu'il est indéfini ou non pertinent, l'action est dite *atélique*. Une autre propriété aspectuelle est la durée qui distingue les verbes d'état dits *statif* des actions dites

duratives, indépendamment de l'existence d'un point final perçu ou non. Krifka (1998) a établit que la télicité est une propriété de l'ensemble du syntagme verbale et n'est pas une caractéristique du verbe seul. En outre, le contexte est un autre facteur qui détermine la classe aspectuelle d'un syntagme verbal (Siegel, 1998). La télicité n'est donc pas une propriété facile à estimer, en particulier dans les langues qui ont une morphologie flexionnelle pauvre comme l'anglais. Elle n'en demeure pas moins indispensable pour de nombreuses tâches de TAL. L'aspect fournit notamment des informations sur les relations temporelles (Costa & Branco, 2012), sur l'implication textuelle (Hosseini *et al.*, 2018; Kober *et al.*, 2019) et l'ordonnancement des événements (Chambers *et al.*, 2014).

Dans cet article, nous montrons que les architectures de type *transformers* sont capables de déterminer la télicité et la durée lorsqu'on leur applique un *fine-tuning*. L'entraînement est réalisé au moyen d'un jeu de données fournies par Friedrich & Gateva (2017) et des versions pré-entraînées de plusieurs modèles TRANSFORMERS mis à disposition sur Huggingface (Wolf *et al.*, 2020). L'évaluation des modèles est quantitative (jeu de test issu du jeu de données) et qualitative (ensemble de phrases simples et de paires minimales). Les résultats obtenus montrent que les modèles BERT sont les plus performants dans la classification binaire de la télicité et de la durée ; les scores RoBERTa, XLNet et ALBERT sont à l'inverse les plus faibles.

2 État de l'art

Siegel & McKeown (2000) ont mis au point plusieurs méthodes de classification aspectuelle fondés sur l'identification de marqueurs linguistiques et ont observé que les méthodes par apprentissage supervisé permettent d'obtenir les meilleurs résultats. Friedrich & Palmer (2014) utilisent pour leur part une approche semi-supervisée d'apprentissage de l'aspect lexical, combinant des caractéristiques linguistiques et distributionnelles, afin de prédire la stativité et la durée. Friedrich & Pinkal (2015) reprennent la même approche pour classer l'aspect lexical verbal en plusieurs catégories duratives (habituel, épisodique, statique) puis Friedrich *et al.* (2016) étendent leurs jeux de données et leurs catégories et atteignent une précision de 76% pour la classification supervisée, se rapprochant ainsi des performances humaines estimées à 80%. Plus récemment, Friedrich & Gateva (2017) font état d'une amélioration significative de la classification automatique de la télicité avec un modèle de régression logistique supervisée.

Loáiciga & Grisot (2016) exploitent la télicité pour améliorer la traduction automatique français-anglais. Falk & Martin (2016) prédisent l'aspect verbal dans différents types de contexte par des méthodes d'apprentissage automatique. Pour leur part, Peng (2018) utilise deux modèles compositionnels PLF et LSA pour classer l'aspect en considérant l'ensemble de la proposition et pas seulement le verbe, sans recourir à des données annotées. L'auteur met en évidence l'importance du syntagme verbale et des dépendants du verbe dans l'interprétation de la télicité. Kober *et al.* (2020) utilisent des modèles distributionnels compositionnels pour déterminer l'aspect des verbes anglais en contexte. Leur travail confirme que le contexte du verbe et les mots grammaticaux qui expriment le temps sont des caractéristiques déterminantes pour la classification aspectuelle.

3 Expériences

Nos expérimentations sont basées sur le *fine-tuning* de modèles *transformers* pour classer des séquences relativement à leur télicité et leur durée (séparément). L'exactitude des modèles spécialisés (*fine-tuned*) est testée en prédisant la télicité et la durée de phrases annotées manuellement. Le *fine-tuning* est une méthode qui consiste à adapter un modèle à une tâche spécifique, en ajoutant une couche supplémentaire dédiée à la tâche en question. Il est ainsi possible d'exploiter les connaissances existantes du modèle et de le spécialiser sur une tâche spécifique sans disposer d'importantes ressources spécialisées, sans avoir recours à une grande puissance de calcul ni à un entraînement long.

Les annotations de télicité et de durée que nous utilisons étant basées sur le verbe principal de la phrase, nous affinons chaque modèle de deux manières : (*i*) en l'entraînant uniquement sur les entrées et les étiquettes de télicité ou de durée ; (*ii*) en fournissant en plus un vecteur `token_type_ids` qui indique la position du verbe dans la séquence d'entrée comme illustré en (1).

(1)

tokens	He	**worked**	well	and	earned	much	.	[SEP]
token_type_ids	0	1	0	0	0	0	0	0

Les modèles sont affinés en utilisant les jeux de données *gold* et *silver* développés et distribués par Friedrich & Gateva (2017). Les annotations *gold* sont basées sur le jeu de données MASC (Ide *et al.*, 2008), tandis que les annotations *silver* ont été construites en utilisant le corpus parallèle InterCorp anglais-tchèque (Čermák & Rosen, 2012) qui permet de déterminer la télicité et la durée des phrases anglaise en exploitant les marqueurs morphologiques dans les phrases tchèques correspondantes. Nous avons extrait 6 354 phrases annotées pour la télicité (3 220 téliques, 3 134 atéliques) et 5 119 phrases pour la durée (1 861 statiques, 3 258 dynamiques) des jeux de données de Friedrich & Gateva. Nous avons par ailleurs augmenté le nombre d'annotations initiales en considérant que les phrases annotées comme statives (pour la durée) sont atéliques, et que les phrases annotées comme duratives sont téliques. Les phrases ont été pré-traitées : séparation des tokens, conversion en minuscule, troncation à 128 mots, remplissage (*padding*), comme cela est recommandé pour le *fine-tuning*. La troncation n'a pas posé de problèmes car une seule phrase dépasse ce seuil et parce que le verbe cible se trouve toujours dans les 128 premiers mots. Afin de mener une évaluation qualitative, nous avons préparé un deuxième ensemble de données de test composé de 40 phrases annotées pour la télicité et 40 phrases annotées pour la durée, réparties de manière égale dans les quatre catégories télique, atélique, statique et durative. Nous avons également construit un 3e jeu de données composé de « paires minimales » de phrases téliques et atéliques qui soit partagent le même verbe soit ne diffèrent que par le verbe (2).

(2)

télique : The girl **walked** a kilometer yesterday.	atélique : The girl **walked** yesterday.
télique : She **noticed** him.	atélique : She **looked** at him.

Nous utilisons des modèles pré-entraînés de la bibliothèque Python `transformers` (Wolf *et al.*, 2020), et en particulier les modèles de classification de séquences. La mise en œuvre reprend les recommandations de l'équipe qui a développée la bibliothèque de *fine-tuning* de ces modèles. Les architectures utilisées sont : BERT, RoBERTa, XLNet et ALBERT, dans les versions de `base`, `large` (taille grande), `cased` (avec les majuscules) et `uncased` (en minuscules) disponibles.

En outre, deux autres modèles de classification binaire sont utilisés comme méthodes de base : un modèle de régression logistique simple implémenté en Python (Celik, 2021) avec les paramètres par défaut, une lemmatisation et la suppression des mots vides et un modèle à réseau de neurones convolutionnel (CNN) implémenté avec Keras (Schapira, 2019). Ce deuxième modèle est largement utilisé comme méthode de base pour les tâches de classification de texte (Kim, 2014).

4 Résultats

4.1 Évaluation quantificative

Au cours du *fine-tuning*, nous avons déterminé les performances des modèles dans la prédiction des étiquettes binaires sur un jeu de données de validation (10% des phrases). La précision et la matrice de confusion ont été calculées en utilisant la bibliothèque Python `scikit-learn` (Pedregosa *et al.*, 2011). Les résultats pour les données de test (10% des phrases) sont présentés en Table 1 pour la télicité et en Table 2 pour la durée. Les modèles les plus performants pour la **télicité** sont `bert-base-cased` et `bert-large-cased`. Globalement, les modèles BERT sont significativement meilleurs que les autres architectures, les modèles RoBERTa étant modérément performants tandis que XLNet et ALBERT (`xlnet-large-cased`, `albert-base-v2`, `albert-large-v2`) tendent à prédire la même étiquette pour toutes les phrases. Par ailleurs, on observe une amélioration nette pour les modèles les plus performants lorsque la position du verbe dans la phrase est fournie lors de l'entraînement. La précision augmente par exemple de 11% pour `bert-base-cased` (65% → 76%) et pour `bert-large-cased` (68% → 79%) . En revanche, elle baisse pour les modèles les moins performants. Les modèles qui obtiennent les meilleurs résultats surpassent aussi très largement nos deux méthodes de base (CNN et régression logistique), les performances de ces dernières s'avérant proches de celles des modèles les moins performants.

						télique			atélique		
modèle	**posit. verbe**	**exact.**	**précis.**	**rappel**	**F1-score**	**précis.**	**rappel**	**F1-score**	**précis.**	**rappel**	**F1-score**
bert-base-uncased	oui	0.72	0.72	0.72	0.72	0.73	0.71	0.72	0.72	0.69	0.71
	non	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.65	0.65	0.65
bert-base-cased	oui	0.76	0.76	0.76	0.76	0.75	0.79	0.77	0.77	0.73	0.75
	non	0.65	0.65	0.65	0.65	0.67	0.61	0.64	0.63	0.69	0.66
bert-large-uncased	oui	0.64	0.64	0.64	0.64	0.66	0.61	0.63	0.63	0.67	0.65
	non	0.66	0.66	0.66	0.66	0.67	0.64	0.65	0.65	0.67	0.66
bert-large-cased	oui	**0.79**	**0.79**	**0.79**	**0.79**	**0.8**	**0.8**	**0.8**	**0.79**	**0.79**	**0.79**
	non	0.68	0.68	0.68	0.68	0.69	0.67	0.68	0.67	0.69	0.68
roberta-base	non	0.64	0.64	0.64	0.64	0.64	0.67	0.65	0.64	0.61	0.63
roberta-large	non	0.66	0.66	0.66	0.66	0.66	0.70	0.68	0.67	0.62	0.64
xlnet-base-cased	oui	0.59	0.59	0.59	0.59	0.59	0.62	0.61	0.59	0.56	0.58
	non	0.61	0.61	0.61	0.61	0.62	0.59	0.60	0.60	0.62	0.61
xlnet-large-cased	oui	0.59	0.59	0.59	0.59	0.59	0.62	0.61	0.59	0.56	0.58
	non	0.51	0.26	0.51	0.34	0.51	1.00	0.67	0.00	0.00	0.00
albert-base-v2	oui	0.49	0.24	0.49	0.33	0.00	0.00	0.00	0.49	1.00	0.66
	non	0.60	0.60	0.60	0.60	0.61	0.60	0.61	0.60	0.60	0.60
albert-large-v2	oui	0.49	0.24	0.49	0.33	0.00	0.00	0.00	0.49	1.00	0.66
	non	0.49	0.24	0.49	0.33	0.00	0.00	0.00	0.49	1.00	0.66
CNN (50 epochs)	non	0.6	0.6	0.6	0.6	0.6	0.62	0.61	0.6	0.58	0.59
Régression logistique	non	0.53	0.63	0.53	0.42	0.52	0.97	0.68	0.74	0.08	0.15

TABLE 1 – Résultats pour la classification de la télicité pour le jeu de données de Friedrich & Gateva.

Les résultats obtenus pour la classification de la **durée** sont globalement meilleurs que ceux de la télicité bien que le jeu de données soit déséquilibré et plus petit. Les modèles `bert-base` l'emportent sur les modèles `bert-large`, tandis que les modèles `roberta-large`, `xlnet-large-case` et `albert-large-v2` s'avèrent incapables de réaliser cette tâche. Nous constatons à nouveau une amélioration significative de la précision lorsque les informations sur la position du verbe sont fournies aux modèles, en particulier pour les plus performants : 70% → 86% pour `bert-base-cased`, 71% → 86% pour `bert-base-uncased`. Là encore, les deux méthodes de base obtiennent des résultats nettement inférieurs à ceux des meilleurs modèles, du même ordre que ceux des modèles

modèle	posit. verbe	exact.	précis.	rappel	F1-score	stative précis.	stative rappel	stative F1-score	durative précis.	durative rappel	durative F1-score
bert-base-uncased	oui	0.86	0.85	0.86	0.85	0.83	0.76	0.79	0.87	0.91	0.89
	non	0.71	0.71	0.71	0.71	0.62	0.57	0.59	0.76	0.80	0.78
bert-base-cased	oui	**0.86**	**0.86**	**0.86**	**0.86**	**0.82**	**0.8**	**0.81**	**0.89**	**0.9**	**0.89**
	non	0.70	0.70	0.70	0.70	0.59	0.55	0.57	0.75	0.78	0.77
bert-large-uncased	oui	0.77	0.77	0.77	0.77	0.70	0.65	0.67	0.81	0.84	0.82
	non	0.70	0.69	0.70	0.70	0.6	0.53	0.56	0.75	0.79	0.77
bert-large-cased	oui	0.74	0.73	0.74	0.73	0.66	0.58	0.62	0.78	0.83	0.80
	non	0.71	0.71	0.71	0.71	0.6	0.58	0.59	0.77	0.78	0.77
roberta-base	non	0.72	0.71	0.72	0.71	0.65	0.49	0.56	0.74	0.85	0.79
roberta-large	non	0.64	0.41	0.64	0.50	0.00	0.00	0.00	0.64	1.00	0.78
xlnet-base-cased	oui	0.70	0.69	0.70	0.68	0.65	0.39	0.49	0.72	0.88	0.79
	non	0.71	0.70	0.71	0.69	0.65	0.43	0.52	0.73	0.87	0.79
xlnet-large-cased	oui	0.64	0.41	0.64	0.50	0.00	0.00	0.00	0.64	1.00	0.78
	non	0.64	0.41	0.64	0.50	0.00	0.00	0.00	0.64	1.00	0.78
albert-base-v2	oui	0.80	0.80	0.80	0.78	0.84	0.54	0.66	0.78	0.94	0.86
	non	0.68	0.66	0.68	0.66	0.59	0.37	0.46	0.70	0.86	0.77
albert-large-v2	oui	0.64	0.41	0.64	0.50	0.00	0.00	0.00	0.64	1.00	0.78
	non	0.64	0.41	0.64	0.50	0.00	0.00	0.00	0.64	1.00	0.78
CNN (50 epochs)	non	0.65	0.64	0.65	0.64	0.54	0.39	0.45	0.7	0.81	0.75
Régression logistique	non	0.64	0.41	0.64	0.50	0.00	0.00	0.00	0.64	1.00	0.78

TABLE 2 – Résultats pour la classification de la durée pour le jeu de données de Friedrich & Gateva.

sous-performants.

4.2 Évaluation qualitative

Les modèles ont également été testés sur deux jeux de données plus petits que nous avons créés et annotés pour la télicité et la durée afin de tester l'exactitude de la classification sur des phrases qui ne proviennent pas du jeu de données de Friedrich & Gateva.

La précision obtenue avec le premier jeu est meilleure que celle obtenue pour les jeux de test originaux. Elle est notamment nettement plus élevée pour les modèles les plus performants : pour la **télicité**, la précision de `bert-base-uncased` la plus élevée est de 75% sans les positions des verbes et de 85% avec ; `bert-large-cased` s'avère en revanche moins efficace avec une précision de 69% sans les positions des verbes et 70% avec. Nous avons également réalisé une évaluation plus qualitative en examinant les mauvaises prédictions des modèles BERT, ces modèles étant globalement les plus performants. Pour presque tous, les erreurs de classification concernent seulement certaines phrases spécifiques dans lesquelles le syntagme verbal définit un aspect temporel inverse de celui qui est spécifié par une partie du contexte : un syntagme prépositionnel comme dans *I eat a fish for lunch on Fridays* ou le temps grammatical comme dans *The inspectors are always checking every document very carefully*. Dans ces deux exemples, l'action est perçue comme ayant un point final mais le temps continu et la présence de l'adverbe *always* rendent la phrase atélique.

Pour la classification de la **durée**, les résultats du premier jeu de données sont encore meilleurs, avec `bert-base-cased` atteignant une précision de 98% (et 92% sans les vecteurs de position) et `bert-large-cased` une précision de 95% avec ou sans vecteurs de position. Cette amélioration n'est pas surprenante, les modèles étant tous plus performants sur la tâche de classification de la durée, mais aussi parce qu'il est difficile de construire des phrases dans lesquelles le contexte et le verbe expriment des valeurs de durée inverses. La phrase *durative* qui a été le plus souvent mal classée par les modèles est *She's playing tennis right now* ; cette erreur est inattendu, car *play* est toujours un verbe d'action. À l'inverse, *Do you hear music ?* est classée comme *durative* par certains modèles parce qu'ils n'ont probablement pas réussi à capter les connaissances du monde nécessaires à son

interprétation (Rogers *et al.*, 2021).

Les résultats des tests que nous venons de présenter montrent tous que les modèles BERT sont les plus performants. Cependant, certaines questions restent sans réponse. Nous avons donc réalisé des tests supplémentaires sur les modèles de classification au moyen du deuxième jeu de test composé de couples de phrases téliques et atéliques qui partagent le même verbe. Le modèle `bert-base-uncased` obtient les meilleurs résultats avec une précision de 81% pour la classification de la télicité lorsque la position des verbes est fournie. En examinant les mauvaises prédictions, nous observons que certaines phrases sont mal classées par tous les modèles. Ce résultat est attendu, car les paires minimales ont des valeurs de télicité opposées tout en ayant le même verbe et parce que les verbes présentent des affinités fortes avec l'une ou l'autre de ces valeurs. Par exemple, la phrase *The boy is eating an apple* est considérée comme télique, car l'action a un terme perçu (l'objet étant au singulier, l'action du verbe est télique) mais la présence d'un temps continu conduit les modèles à classer incorrectement la phrase comme atélique. De même, la phrase *The Prime Minister made that declaration for months* serait télique, sans la présence du complément de temps *for months*. Ces exemples suggèrent que les modèles accordent trop d'importance au verbe et ne tiennent pas suffisamment compte du temps grammatical et du contexte, notamment lorsque ce dernier contient des compléments de temps.

Afin de tester davantage encore les modèles, nous avons étudié l'effet d'un masque d'attention sur le contexte et la façon dont il affecte la classification. Cette méthode a été notamment utilisée par Metheniti *et al.* (2020). Les masques d'attention imposent aux modèles de prédire la télicité ou la durée en ne considérant que le verbe. Ils ont été appliqués aux phrases d'entrée lors de la phase de test, sans procéder à un nouveau *fine-tuning* des modèles. La capacité des modèles à déterminer la télicité diminue de manière significative, lorsque la prédiction est réalisée uniquement sur la base du verbe, $79\% \rightarrow 51\%$ par exemple pour `bert-large-cased` sur le jeu de données de Friedrich & Gateva. La baisse rend compte de l'importance du contexte et des dépendants du verbe dans la prédiction de la télicité et de la durée. Ces résultats sont conformes aux prédictions de théories linguistiques comme celle de Krifka (1998) : l'aspect ne dépend pas uniquement du verbe ; il est également déterminé par le contexte.

5 Discussion

Le *fine-tuning* des modèles *transformers* produit des résultats à l'état de l'art pour de nombreuses applications de TALN et dans de nombreux domaines. Ces modèles font cependant l'objet de critiques car il s'agir de « boîte noire » tant au niveau de leur création que de leur déploiement. Les critiques concernent également les stratégies sous-optimales de *fine-tuning* devenues courantes. Dodge *et al.* (2020) soulignent notamment que l'initialisation du processus de *fine-tuning* avec une amorce aléatoire peut produire des résultats sensiblement différents, même avec les mêmes hyperparamètres.

Bien que nos jeux de données soient relativement petits (6K pour la télicité et 4K pour la durée), nous ne les avons pas mélangés avant de les diviser en données d'entraînement, de test et de validation suivant en cela la proposition de Dodge *et al.* (2020). Par ailleurs, nous avons utilisé l'optimisateur ADAM de PyTorch comme cela est recommandé par Zhang *et al.* (2020) au lieu de BERTADAM (Devlin *et al.*, 2019; Wolf *et al.*, 2020). À l'inverse, nous avons suivi les recommandations de Devlin *et al.* (2019) et McCormick & Ryan (2019) de réduire le nombre d'époques d'entraînement et de choisir la meilleure époque en fonction des résultats de validation, la proposition de Dodge *et al.*

(2020) et Mosbach *et al.* (2020) de multiplier les époques d'entraînement pour toutes les tâches s'avérant en définitive contre-productive (Zhang *et al.*, 2020). Signalons également que nous avons répété l'entraînement des modèles avec 75%, 80% et 90% des jeux de données sans observer aucune différence significative dans leur comportement ni aucune baisse de performance.

Notons également que nous nous attendions à ce que les modèles RoBERTa soient moins performants sur nos tâches, car ils exploitent mal les informations contextuelles de mots entiers et ne tiennent pas suffisamment compte des vecteurs de position des verbes. Cependant, nous ne nous attendions pas à ce que les modèles XLNet et ALBERT soient aussi peu performants. XLNet dépasse en effet les modèles BERT sur les tâches de la compréhension et de la classification des textes (Yang *et al.*, 2019) ; ALBERT dépasse BERT lui aussi sur plusieurs tâches, l'une des innovations de cette architecture étant justement ses représentations dépendantes du contexte (Wright, 2019). XLNet et ALBERT obtiennent globalement de mauvais résultats dans toutes les expériences que nous avons réalisées. Cela suggère qu'ils ne sont pas adaptés au *fine-tuning* au moyen de jeux de d'entraînement de petite taille ni à la classification binaire de séquences courtes, ces dernières ne leurs permettant pas de tirer profit de leurs points forts comme la meilleure prise en compte des relations à longue distance.

Une autre question intéressante qui émerge de nos expériences concerne le succès des versions `cased` de BERT par rapport aux versions `uncased`. Pourquoi les modèles qui conservent les majuscules sont-ils plus performants alors que nos jeux de données sont entièrement en minuscules ? Rappelons que les modèles où les mots conservent leurs majuscules sont essentiellement utilisés pour la reconnaissance d'entités nommées pour laquelle ces informations supplémentaires sont importantes. En outre, la version `large-cased` de BERT donne les meilleurs résultats dans la plupart des cas, ou est à égalité avec les versions de `base`. Or nous savons que les gros modèles *transformers* sont plus difficiles à ajuster, surtout lorsque les jeux de données sont petits. Les bons résultats de `bert-large-cased` s'expliquent probablement par l'homogénéité relative de nos jeux de données composés de phrases de genres similaires (littérature, articles de presse).

6 Conclusion

Nous avons mené dans cette étude plusieurs expériences qui testent la capacité des modèles *transformers* à capter les catégories aspectuelles comme la télicité et la durée. Nous avons testé cette capacité en réalisant un *fine-tuning* de modèles de classification binaire au moyen notamment du jeu de données annotées pour la télicité et la durée de (Friedrich & Gateva, 2017). Le *fine-tuning* a été réalisé sur des modèles *transformers* de plusieurs architectures (BERT, RoBERTa, XLNet, ALBERT). Nous avons ainsi observé que malgré la taille réduite de nos jeux de données, certains modèles sont très efficaces pour la classification aspectuelle et que les performances sont considérablement améliorées lorsque la position du verbe dans la phrase est fournie au classifieur lors de l'entraînement. L'examen des erreurs des classifieurs nous a permis de caractériser les limites des modèles, notamment pour les phrases où l'information temporelle exprimée dans le contexte est à rebours de l'aspect verbal. Enfin, nous avons mis en évidence l'importance du contexte dans la prédiction de l'aspect en utilisant des masques d'attention.

Références

CELIK I. (2021). Text classification logistic regression from scratch. github.com/iremcelik/Text-Classification-Logistic-Regression-From-Scratch.

CHAMBERS N., CASSIDY T., MCDOWELL B. & BETHARD S. (2014). Dense Event Ordering with a Multi-Pass Architecture. In *Transactions of the Association for Computational Linguistics*, volume 2, p. 273–284. DOI : 10.1162/tacl_a_00182.

COSTA F. & BRANCO A. (2012). Aspectual Type and Temporal Relation Classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 266–275, Avignon, France : Association for Computational Linguistics.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DODGE J., ILHARCO G., SCHWARTZ R., FARHADI A., HAJISHIRZI H. & SMITH N. (2020). Fine-tuning pretrained language models : Weight initializations, data orders, and early stopping. *arXiv preprint arXiv :2002.06305*.

DOWTY D. R. (1979). *Word Meaning and Montague Grammar : The Semantics of Verbs and Times in Generative Semantics and in Montague's Ptq*, volume 7. Springer.

FALK I. & MARTIN F. (2016). Automatic identification of aspectual classes across verbal readings. In * *Sem 2016 THE FIFTH JOINT CONFERENCE ON LEXICAL AND COMPUTATIONAL SEMANTICS*.

FRIEDRICH A. & GATEVA D. (2017). Classification of telicity using cross-linguistic annotation projection. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 2559–2565.

FRIEDRICH A. & PALMER A. (2014). Automatic prediction of aspectual class of verbs in context. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 517–523.

FRIEDRICH A., PALMER A. & PINKAL M. (2016). Situation entity types : automatic classification of clause-level aspect. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1757–1768.

FRIEDRICH A. & PINKAL M. (2015). Automatic recognition of habituals : a three-way classification of clausal aspect. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 2471–2481.

HOSSEINI M. J., CHAMBERS N., REDDY S., HOLT X. R., COHEN S. B., JOHNSON M. & STEEDMAN M. (2018). Learning Typed Entailment Graphs with Global Soft Constraints. In *Transactions of the Association for Computational Linguistics*, volume 6, p. 703–717. DOI : 10.1162/tacl_a_00250.

IDE N., BAKER C., FELLBAUM C., FILLMORE C. & PASSONNEAU R. (2008). MASC : the Manually Annotated Sub-Corpus of American English. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco : European Language Resources Association (ELRA).

KIM Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, **abs/1408.5882**.

KOBER T., ALIKHANI M., STONE M. & STEEDMAN M. (2020). Aspectuality Across Genre : A Distributional Semantics Approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4546–4562, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.401.

KOBER T., BIJL DE VROE S. & STEEDMAN M. (2019). Temporal and Aspectual Entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, p. 103–119, Gothenburg, Sweden : Association for Computational Linguistics. DOI : 10.18653/v1/W19-0409.

KRIFKA M. (1998). The origins of telicity. In *Events and grammar*, p. 197–235 : Springer.

LOÁICIGA S. & GRISOT C. (2016). Predicting and Using a Pragmatic Component of Lexical Aspect of Simple Past Verbal Tenses for Improving english-to-french Machine Translation. In *Linguistic Issues in Language Technology, Volume 13, 2016* : CSLI Publications.

MCCORMICK C. & RYAN N. (2019). BERT Fine-Tuning Tutorial with PyTorch. Retrieved January 24, 2021.

METHENITI E., VAN DE CRUYS T. & HATHOUT N. (2020). How Relevant Are Selectional Preferences for Transformer-based Language Models ? In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 1266–1278, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.109.

MOSBACH M., ANDRIUSHCHENKO M. & KLAKOW D. (2020). On the stability of fine-tuning BERT : Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv :2006.04884*.

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 : Nachine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

PENG Q. (2018). *Towards aspectual classification of clauses in a large single-domain corpus*. Edingburgh, UK : School of Informatics, University of Edinburgh.

ROGERS A., KOVALEVA O. & RUMSHISKY A. (2021). A Primer in BERTology : What we know about how BERT works. In *Transactions of the Association for Computational Linguistics*, volume 8, p. 842–866 : MIT Press.

SCHAPIRA D. (2019). diegoschapira/cnn-text-classifier-using-keras.

SIEGEL E. V. (1998). *Linguistic Indicators for Language Understanding : Using machine learning methods to combine corpus-based indicators for aspectual classification of clauses*. Columbia University. Ph.D. thesis.

SIEGEL E. V. & MCKEOWN K. R. (2000). Learning Methods to Combine Linguistic Indicators :Improving Aspectual Classification and Revealing Linguistic Insights. In *Computational Linguistics*, volume 26, p. 595–627.

VERKUYL H. J. (1972). *On the compositional nature of the aspects*, volume 15. Springer.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). Transformers : State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 38–45, Online.

WRIGHT L. (2019). Meet ALBERT : a new 'Lite BERT' from Google & Toyota with State of the Art NLP performance and 18x fewer parameters. Retrieved January 24, 2021.

YANG Z., DAI Z., YANG Y., CARBONELL J., SALAKHUTDINOV R. R. & LE Q. V. (2019). XLNet : Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, **32**, 5753–5763.

ZHANG T., WU F., KATIYAR A., WEINBERGER K. Q. & ARTZI Y. (2020). Revisiting few-sample BERT fine-tuning. *arXiv preprint arXiv :2006.05987*.

ČERMÁK F. & ROSEN A. (2012). The Case of InterCorp, a multilingual parallel corpus. In *International Journal of Corpus Linguistics*, volume 13, p. 411–427. DOI : 10.1075/ijcl.17.3.05cer.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Sifting French Tweets to Investigate the Impact of Covid-19 in Triggering Intense Anxiety

Mohamed-Amine Romdhane, Elena Cabrio and Serena Villata
Université Côte d'Azur, Inria, CNRS, I3S, France
mohamed-amine.romdhane@etu.univ-cotedazur.fr,
elena.cabrio@univ-cotedazur.fr, villata@i3s.unice.fr

RÉSUMÉ

Les réseaux sociaux peuvent être exploités pour comprendre les sentiments et les émotions des personnes en temps réel et cibler les messages de santé publique en fonction de l'intérêt et des émotions des utilisateurs. Dans cet article, nous étudions l'impact de la pandémie COVID-19 dans le déclenchement des crises d'angoisse, en nous appuyant sur les messages échangés sur Twitter. Plus précisément, nous fournissons : *i)* une analyse quantitative et qualitative d'un corpus de tweets en français liés au coronavirus, et *ii)* une approche en pipeline (un mécanisme de filtrage suivi par des méthodes de réseaux de neurones) pour classer de manière satisfaisante les messages exprimant de l'anxiété sur les médias sociaux, en considérant le rôle joué par les émotions.

ABSTRACT

Sifting French Tweets to Investigate the Impact of Covid-19 in Triggering Intense Anxiety.

Social media can be leveraged to understand public sentiment and feelings in real-time, and target public health messages based on user interests and emotions. In this paper, we investigate the impact of the COVID-19 pandemic in triggering intense anxiety, relying on messages exchanged on Twitter. More specifically, we provide : *i)* a quantitative and qualitative analysis of a corpus of tweets in French related to coronavirus, and *ii)* a pipeline approach (a filtering mechanism followed by Neural Network methods) to satisfactory classify messages expressing intense anxiety on social media, considering the role played by emotions.

MOTS-CLÉS : détection de l'anxiété, COVID-19, données Twitter, apprentissage automatique, apprentissage profond.

KEYWORDS: intense anxiety detection, COVID-19, Twitter data, machine learning, deep learning.

1 Introduction

The COVID-19 pandemic - that overtook most of the world's countries - is forcing government-issued lockdowns, strict hygiene regulations and is ultimately causing global panic, uncertainty and fear. During this crisis, social media are representing a valuable source of news and a medium for expressing people's opinions and sentiment about the emergency [1] and the restrictive measures deployed by the different countries to fight COVID-19 spread. Given that social media are proving instrumental in keeping people connected during the crisis, they turned out to be beneficial for mental

1. https://www.medrxiv.org/content/10.1101/2020.04.03.20052936v1

health, in helping to combat widespread feelings of loneliness stemming from extended periods of isolation and social distancing. However, panic attacks are reported to be increasing during COVID-19, as people are increasingly worried about their health [2]. Panic attacks are characterized by an intense fear and sense of feeling overwhelmed, according to the National Institutes of Mental Health. Fear of contamination, sleep disturbance and probability of an economic slowdown with potential job losses are among the major factors leading to depression and anxiety among people (as emerges from the following tweets : *"Coucou...insomnie depuis 15 jours....merci corona."(EN : "Hi... insomnia since 15 days... thanks corona"* ; *"Coronavirus au début jmen foutai de oufff mais mntn vazy y commence a être chaud" (EN : Coronavirus at the beginning I was not caring but now it's heated)*). Despite how scary they can feel, anxiety attacks are relatively common, and in most of the cases feelings are manageable. However, if multiple anxiety attacks happen, or fear over having a panic attack gets in the way, this may be a sign of anxiety disorder and a person should seek help from a mental health professional.

Starting from these considerations and from the observation that people are heavily using social media to express - among others - their feelings about the current sanitary situation, the goal of the current work is to investigate the impact of the COVID-19 pandemic in triggering intense anxiety, relying on messages exchanged on Twitter. Such research issue breaks down into the following research questions : *Can we automatically distinguish tweets expressing a person's intense anxiety status from tweets that are more factual or expressing general feelings on COVID-19 ?* and *Does emotion detection in tweets help improving such task ?* To address such research questions, we analyse a corpus of tweets in French, with the goal of delivering automated methods to detect severe anxiety in social media messages. As a first step, we propose a qualitative and quantitative study on a subset of French tweets of the multilingual Twitter COVID-19 Dataset (Chen *et al.*, 2020). We carried out an annotation process to classify such tweets as expressing severe anxiety, or not. Moreover, we also annotated each tweets with one (or more) emotion(s) (Ekman, 1992) and their intensity, to investigate the correlations between emotions and anxiety. We then propose and experiment with a pipeline approach (keyword-based filtering + Neural Network models) to classify such tweets as containing or not severe anxiety, relying on a set of features including emotions, obtaining satisfactory results. Our findings, together with other analysis as the monitoring of Google Trends can provide continued surveillance and guide public mental health initiatives to mitigate the psychological toll of COVID-19.

2 Anxiety-COVID19 Dataset

For our study we rely on a subset of the French tweets of the multilingual Twitter COVID-19 Dataset (Chen *et al.*, 2020), a large-scale public Twitter dataset. Such ongoing data collection started in January 2020 by tracking COVID-19-related keywords and accounts [3] in multiple languages.

Data collection and annotation. From the COVID-19 Dataset, we extract only the tweets in French posted from March to May 2020 (2.7 million tweets). As only a very small percentage of tweets is actually written by people expressing severe anxiety, we decided to apply a filtering strategy to narrow down our search space to those tweets in the corpus expressing worries and troubles related to coronavirus. As a first step, a linguist helped us in compiling a list of keywords, i.e., unigrams

2. Since the beginning of the pandemics, Google Trends has revealed a massive uptick in the rise of searches related to anxiety, panic attacks, and treatments for panic attacks https://www.weforum.org/agenda/2020/09/google-trends-panic-attack-anxiety-self-help-rise-covid

3. https://github.com/echen102/COVID-19-TweetIDs

Formal language	Colloquialism
inquiet, inquietant, inquietude ; stress, stressé, stressant ; anxieux, anxiété ; angoisse, angoissé, angoissant ; terrifié, terrifiant, terrible ; pleure, pleurer ; crier ; hurler ; triste ; tristesse ; pessimiste ; tourmenté ; soucieux, soucis ; craintes, craindre ; malaise ; trouble ; frayeur ; terreur ; peur, peurs ; panique, paniquer ; redouter ; agité ; larmes	*trac ; flipper, flippant ; je suis pas bien, jsuis pas bien, chui pas bien ; c'est chaud, c chaud ; jsuis en pls, chui en pls ; ça craint ; je suis en pls, c'est éclaté, c éclaté, eclate ; c'est claqué, claque ; je suis mort, jsuis mort, chui mort*
EN : worried ; stress, stressed out, stressful ; freaking out ; freaking out, I'm not well ; it's heated ; I am not well ; I am not anxious, anxious ; anguish, distressing ; terrified ; well, not well ; it's heated ; terrifying, terrible ; cry ; yell ; sad ; feel down ; it sucks ; sadness ; pessimistic ; tormented ; worried, worries ; fears, fear ; discomfort ; trouble ; fright ; terror ; I am dead ; afraid ; panic ; restless ; tears	

TABLE 1: Examples of panic-related keywords

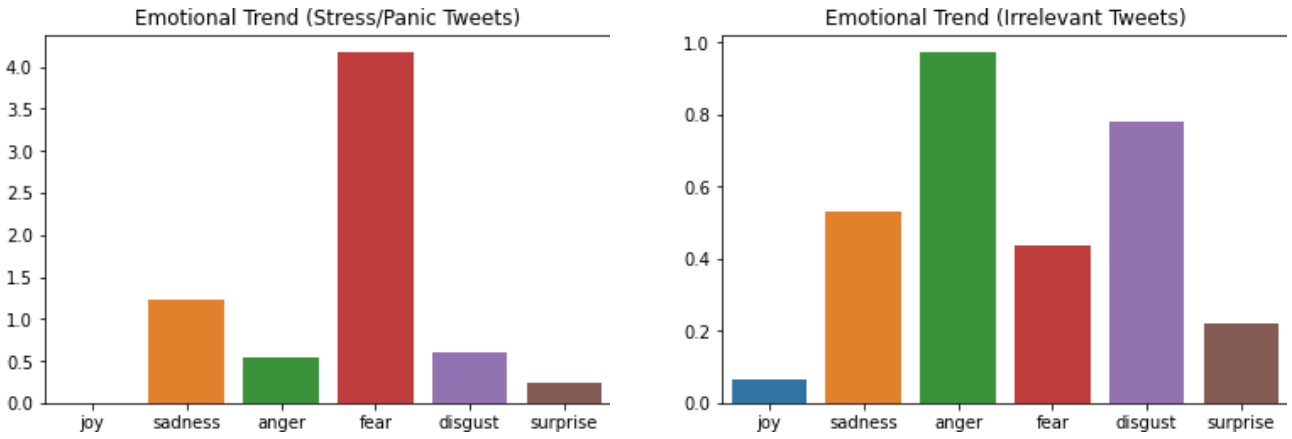
or n-grams, that frequently occur in messages expressing anxiety or stress, considering both formal language and colloquialism (see Table 1). We then apply a string matching algorithm to extract only the tweets containing one or more of those keywords, reducing the initial dataset to ∼33 000 tweets.

As a pilot study, we carry out an annotation process of a sample of 1032 keyword-filtered tweets, to check how many are actually expressing severe anxiety. To create such sample, we selected ∼50 tweets per day starting from the beginning of the lockdown in France in March 2020, till May. We selected this period of time as we aim to study also the correlation between the pandemic evolution and the increase/decrease of messages expressing anxiety about the sanitary situation. Out of the 1032 keyword-filtered and annotated tweets, 114 have been labeled as positive instances of the "anxiety" class (e.g., *"Y'a trop de gens qui je connais qui commencent à mourir du Corona là comment c'est angoissant" (EN : "There are too many people that I know that are dying because of Corona that's frightening")*, *"Le coronavirus il me fait flipper, maintenant tu es enrhumé tu as peur..." (EN : "Coronavirus is driving me crazy now you have a flu and you are scared"))*, while the rest of the tweets is labeled as "non anxiety". In the latter category there are general or factual tweets referring to some news as in *"l'#OMS demande aux personnes de plus de 60 ans et à ceux souffrant d'une maladie respiratoire d'éviter le plus possible de se rendre partout où il y a de la foule." (EN : "The OMS calls on people over 60 and those with respiratory illnesses to avoid traveling to crowded places as much as possible.")*), or tweets containing sarcasm (e.g., *"Maintenant que l'on vas tous mourir du coronavirus mdr il est temps de m'avouer vos sentiments après c'est trop tard" (EN : "Now that all of us will be dying of coronavirus lol I should confess my feelings, after it will be too late")*). To verify the reliability of our annotation, we calculated the inter-annotator agreement (IAA) on a reserved and previously unseen subset of 100 keyword-filtered tweets, sampled randomly from the collected data. Three raters annotated the data independently, resulting in a Fleiss' kappa of 0.67 (meaning substantial agreement).

Following (Li *et al.*, 2020b), we also annotate the same sample of 1032 keyword-filtered tweets with one (or more) emotion(s) (the 6 basic emotions : joy, sadness, anger, fear, disgust and surprise (Ekman, 1992)), and the emotion intensity (a score between 0 and 5), to investigate the correlations between emotions and severe anxiety.

Labels correlation. We calculated the emotions distribution over the keyword-filtered tweets. The emotions of anger, disgust, sadness and fear are the most intense in the keyword-filtered dataset, while joy and surprise have very little to no presence. Anger and disgust arise for unfavourable opinions about the government decisions, e.g., underestimating the scale of the outbreak. The more active cases and deaths were reported in the news, the more the frequency of panic tweets increased. Sadness

generally refers to losses, both in terms of lack of mobility freedom due to lockdown and death tolls.



(a) "Panic" class

(b) "Non panic" class

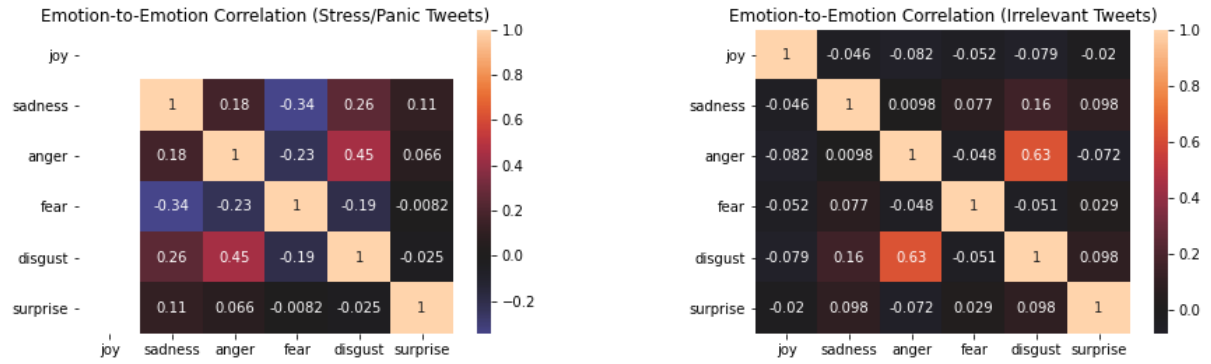
FIGURE 1: Emotions distribution over the keyword-filtered tweets by class

Figures 1a and 1b report on the emotions distribution over the two classes of keyword-filtered tweets. As introduced before, an emotion intensity score ranging from 0 to 5 is provided for the annotated tweets, for each of the 6 basic emotions. Such emotional intensities have been normalized and the average intensity of each emotion is calculated and plotted (Y-axis). Fear is the predominant emotion in the "anxiety" tweets, followed by the other negative emotions and the absence of joy. Anger and disgust dominate the other tweets, followed by sadness. We calculated the emotion-to-emotion correlation using the Pearson correlation coefficient (Figures 2a and 2b). The emotions of anger, disgust and sadness positively correlate to each other, especially due to a person's general feeling of anger or sadness for a particular "disgusting" message or news. The correlation between anger and disgust also seems to occur more in "non anxiety" tweets ; the same positive correlation is found in the "anxiety" tweets coupled with disgust and surprise emotions. Fear has a negative correlation to anger as one overwhelms the other in a given tweet. Moreover, we calculated the correlation between the emotional intensities of a given tweet and the following features : the number of uppercase words, the number of adverbs, the number of unidentified words by the spaCy's PoS Tagger [4], the number of emojis related to each emotion, and the number of retweets. Results on our dataset show that fear positively correlates with the number of emojis expressing it or sadness. Anger is also correlated to the number of adverbs, uppercase words and mentions. Joy, sadness and anger emojis are also present in their respective emotions. As for the tweets expressing anxiety, sadness, anger and disgust positively correlate with a high number of adverbs in the tweet (see Figures 3a and 3b).

3 Classification of intense anxiety messages

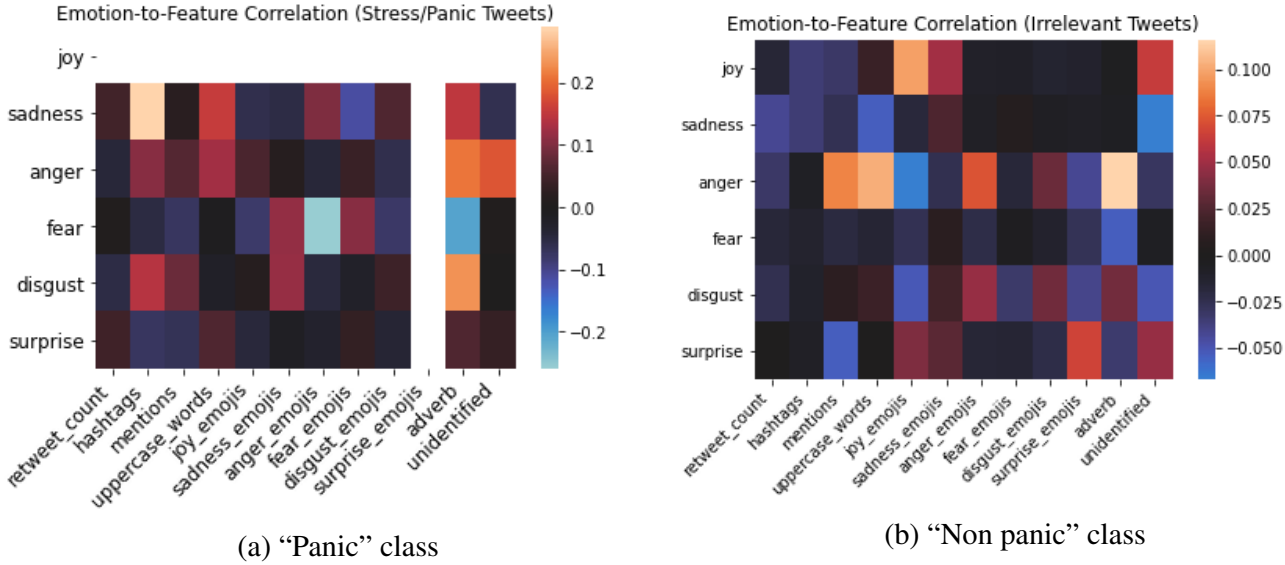
As our goal is to automatically detect messages expressing severe anxiety, we propose a pipeline approach that after employing the filtering mechanisms we described to Twitter messages, applies supervised methods to classify tweets as expressing or not anxiety. We cast this second step as a binary classification task (anxiety tweet / irrelevant), and we experiment over our dataset of 1000 keyword-filtered tweets.

4. https://spacy.io/



(a) “Panic” class (b) “Non panic” class

FIGURE 2: Emotion-to-emotion correlation



(a) “Panic” class (b) “Non panic” class

FIGURE 3: Emotion-to-features correlation

Classification methods. We test the following supervised methods and features :

1. *Bag-of-Words + SVM Classifier*. Baseline method relying on Bag-of-Words, TF-IDF and features from spaCy as input to a Support Vector Machine. Grid Search is performed on the hyperparameters to get the better combination of N-grams, learning rate and loss function.
2. *CamemBERT Embedding + GRU Classifier* : CamemBERT (Martin *et al.*, 2020) is a state-of-the-art language model for French based on the RoBERTa architecture (ready-to-use module in HuggingFace’s (Wolf *et al.*, 2020) “transformers” library).
3. *CamemBERT Embedding + CamemBERT Sequence Classifier* : This model uses a RoBERTa classification head consisting of 2 dense Linear NN layers with a dropout rate of 10%. It is the default Sequence Classifier of CamemBERT.
4. *CamemBERT Embedding (w/ Emotional Embedding) + GRU Classifier* : This model is based on the same CamemBERT model described above (point 2), enhanced by emotion intensities (manually annotated) as features. This combination is made in a rather simple manner where instead of returning the output of the previous model, such output gets fed into a final dense layer where it is processed with the 6 emotion intensities of the current tweet batches.

5. *CamemBERT Embedding (w/ Emotional Embedding) + CamemBERT Sequence Classifier* : same CamemBERT model as described in point 3, plus emotional embeddings as in point 4.

Given the small dataset at our disposal, the evaluation is reported over a 10-Fold Cross-validation (train/validation on 80% of the data, the rest for testing). For the last four models, we perform 10-Fold Cross-validation during a 50 epochs training with early stopping. As a preprocessing step, we get rid of mentions and hashtags at the beginning and end of the tweets. For the last 4 models, we use HuggingFace's CamemBERT Tokenizer to encode tweet token ids and attention masks.

Results and error analysis. Table 2 reports on the obtained results for the binary classification task over the keyword-filtered dataset. Even with a small and unbalanced dataset as ours, state-of-the-art deep neural network models such as CamemBERT obtain promising results (avg. F-measure 0.72), outperforming standard approaches like SVMs. Note that those results are obtained over the keyword-filtered dataset, meaning that if on the one side the minority class ("panic") is more represented than in the COVID-19 dataset thanks to the filtering mechanism, still it represents only the 11% of the dataset. Moreover, all the keyword-filtered messages contain similar terms making it pretty challenging to separate an alarm message from a person (e.g., *"Et mais jvais creuver si j'ai le coronavirus deja jcours 2 mètres au foot je suffoque j'ai besoin de ma ventoline" (EN : "If I have coronavirus I will die, already now when I run 2 meters when playing football I suffocate I need my ventoline")*) from a tweet expressing general worries on the sanitary situation (e.g., *"mais stop créer de la fausse panique y'a déja assez de gens qui over-react avec le Coronavirus" ("EN : Stop generating fake panic there are already too many people that overreact with Coronavirus")*). Therefore, including emotion features helps in improving classification performances on the minority class (for both models). We employed gold-standard emotions as features, but we plan to implement an emotion classifier to extract them automatically.

	Precision		Recall		F-measure	
Models	**Panic**	**Non-Panic**	**Panic**	**Non-Panic**	**Panic**	**Non-Panic**
BOW + SVM	0.40	0.92	0.35	0.93	0.37	0.93
CamemBERT + GRU Classifier	**0.52**	0.94	0.48	**0.95**	0.50	**0.94**
CamemBERT + Sequence Classifier	0.43	0.94	0.52	0.91	0.47	0.93
CamemBERT (w/ Emotional Embedding) + GRU Classifier	0.46	0.94	0.57	0.92	**0.51**	0.93
CamemBERT (w/ Emotional Embedding) + Sequence Classifier	0.39	**0.95**	**0.65**	0.88	0.49	0.91

TABLE 2: Obtained results on the panic messages classification

Concerning the classification errors, for the CamemBERT + GRU Classifier model, tweets such as *"Wallah : Jamais je fais le teste pour le covid en faite :joy_emoji :" (EN : "I will never do the covid test actually :joy_emoji :")* have been annotated as "no anxiety" but predicted otherwise. This may be explained with the model not learning that joy emojis may counter the content of a panic message. Looking to misclassified "anxiety" tweets, tweets like *"J'ai appelé mes parents aujourd'hui. Je suis dépité. [...] Je me suis énervé. Ils n'ont rien compris. Une fois raccroché, j'ai pleurer de colère." (EN : I called my parents today. I'm disappointed. I got angry. They didn't understand anything. When I hung up, I cried in anger.")* are hard for a model with limited anxiety-labeled samples.

Related work. (CALVO *et al.*, 2017; Coppersmith *et al.*, 2018) investigated NLP methods to identify people who may be in need of psychological assistance. (Medford *et al.*, 2020) extract a sample of tweets matching hashtags related to COVID-19 and measure frequency of keywords related to infection prevention practices, vaccination, and racial prejudice. They perform sentiment analysis to identify emotional valence and predominant emotions, and topic modeling to explore discussion topics over time. Similarly, (Xue *et al.*, 2020) and (Sengupta *et al.*, 2020) analyze Twitter messages

related to the COVID-19 pandemic, and apply LDA to identify popular unigrams and bigrams, salient topics and themes, and sentiments in tweets. (Li *et al.*, 2020a) train deep models that classify each tweet into 8 emotions, and build the Emotion-Covid19-Tweet dataset. They investigate the reasons that are causing sadness and fear, and study the emotion trend in both keyword and topic level. While we share the same objective, to the best of our knowledge ours is the first study for French that investigates Twitter messages to unveil the impact of COVID-19 in triggering severe anxiety.

4 Conclusions and Future Work

The main contributions of the paper are *i)* a dataset of 1032 tweets in French annotated with the "anxiety" label and six emotions (despite the small size of the minority class, the collected data are representative, as most of the tweets mention or are related to the major factors mentioned by the medical literature to lead to depression and anxiety among people) ; *ii)* a corpus-based analysis of the correlations between panic messages and emotions, and other linguistic features ; *iii)* a pipeline approach (keyword-based filtering + supervised model) to classify tweets containing panic. As future work, we plan to extend the Anxiety-COVID19 dataset, and to test alternative methods to deal with class imbalance. Moreover, we plan to carry out a study on the impact of the pandemic evolution in time with respect to messages expressing intense anxiety.

Références

CALVO R. A., MILNE D. N., HUSSAIN M. S. & CHRISTENSEN H. (2017). Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, **23**(5), 649–685. DOI : 10.1017/S1351324916000383.

CHEN E., LERMAN K. & FERRARA E. (2020). Tracking social media discourse about the covid-19 pandemic : Development of a public coronavirus twitter data set. *JMIR Public Health Surveill*, **6**(2), e19273. DOI : 10.2196/19273.

COPPERSMITH G., LEARY R., CRUTCHLEY P. & FINE A. (2018). Natural language processing of social media as screening for suicide risk. *Biomedical Informatics Insights*, **10**, 117822261879286. DOI : 10.1177/1178222618792860.

EKMAN P. (1992). An argument for basic emotions. *Cognition and Emotion*, p. 169–200.

LI I., LI Y., LI T., ALVAREZ-NAPAGAO S., GARCIA-GASULLA D. & SUZUMURA T. (2020a). What are we depressed about when we talk about covid-19 : Mental health analysis on tweets using natural language processing. In M. BRAMER & R. ELLIS, Éds., *Artificial Intelligence XXXVII*, p. 358–370, Cham : Springer International Publishing.

LI I., LI Y., LI T., ALVAREZ-NAPAGAO S., GARCIA-GASULLA D. & SUZUMURA T. (2020b). What are we depressed about when we talk about covid19 : Mental health analysis on tweets using natural language processing.

MARTIN L., MÜLLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, p. 7203–7219.

MEDFORD R. J., SALEH S. N., SUMARSONO A., PERL T. M. & LEHMANN C. U. (2020). An "infodemic" : Leveraging high-volume twitter data to understand public sentiment for the covid-19 outbreak. *medRxiv*. DOI : 10.1101/2020.04.03.20052936.

SENGUPTA S., MUGDE S. & SHARMA G. (2020). An exploration of impact of covid 19 on mental health -analysis of tweets using natural language processing techniques. *medRxiv*. DOI : 10.1101/2020.07.30.20165571.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 38–45, Online : Association for Computational Linguistics.

XUE J., CHEN J., HU R., CHEN C., ZHENG C., SU Y. & ZHU T. (2020). Twitter discussions and emotions about the covid-19 pandemic : Machine learning approach. *J Med Internet Res*, **22**(11), e20550. DOI : 10.2196/20550.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Stratégie Multitâche pour la Classification Multiclasse

Houssam Akhmouch[1, 2] Hamza Bouanani[2] Gaël Dias[1] José G. Moreno[3]
(1) Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
(2) Crédit Agricole Brie Picardie, 77100 Meaux, France
(3) Université de Toulouse, IRIT UMR 5505 CNRS, 31000 Toulouse, France
first.last@unicaen.fr, first.last@ca-briepicardie.fr, first.last@irit.fr

RÉSUMÉ

Nous proposons une idée originale pour exploiter les relations entre les classes dans les problèmes multiclasses. Nous définissons deux architectures multitâches de type *one-vs-rest* qui combinent des ensembles de classifieurs appris dans une configuration multitâche en utilisant des réseaux de neurones. Les expériences menées sur six jeux de données pour la classification des sentiments, des émotions, des thématiques et des relations lexico-sémantiques montrent que nos architectures améliorent constamment les performances par rapport aux stratégies de l'état de l'art de type *one-vs-rest* et concurrencent fortement les autres stratégies multiclasses.

ABSTRACT

A Multitask Strategy for Multiclass Classification

We propose an original idea to exploit the relations between classes in multiclass problems. We define two multitask one-vs-rest architectures that combine sets of classifiers learned in a multitask way using neural networks. Experiments over six gold standard data sets for sentiment, emotion, topic and lexico-semantic relations classification show that our architectures steadily improve performance compared to state-of-the-art one-vs-rest strategies, and strongly compete with other multiclass strategies.

MOTS-CLÉS : classification multitâche, classification multiclasse, classification de textes, classification de relations lexico-sémantiques.

KEYWORDS: multitask classification, multiclass classification, text classification, classification of lexico-semantic relations.

1 Introduction

Alors que la classification binaire traite des problèmes à deux classes (par exemple, spam vs. non spam), la classification multiclasse implique l'attribution de plus de deux étiquettes (par exemple, positif, neutre ou négatif). Un large éventail de problèmes multiclasses existe dans le monde réel, tels que l'analyse des sentiments (Balikas *et al.*, 2017), la classification des genres de nouvelles (Wang *et al.*, 2018), la détection des émotions (Ye *et al.*, 2020), pour n'en citer que quelques-uns. Si de nombreux efforts ont été déployés pour (1) affiner des modèles d'apprentissage pour des problèmes spécifiques de classification textuelle (Teng *et al.*, 2016; Zhou & Li, 2020), ou (2) concevoir des architectures (neuronales) spécifiques à la classification de texte (Conneau *et al.*, 2017; Yao *et al.*, 2019; Zhang & Zhang, 2020), moins de travaux se sont attelés à proposer des modèles génériques qui peuvent traiter des problèmes multiclasses indépendamment de la tâche et de l'objet à classer.

Dans ce cadre, trois approches principales ont été proposées pour traiter les problèmes multiclasses. Premièrement, ils peuvent être résolus en étendant des techniques de classification binaire ; directement

pour les arbres de décision, les réseaux bayésiens ou les K plus proches voisins, et avec quelques modifications pour les réseaux de neurones (Ou & Murphey, 2007) et les machines à vecteurs de support (Crammer & Singer, 2001). Deuxièmement, des stratégies ont été développées qui reposent sur une décomposition du problème en un ensemble de sous-problèmes binaires, dont les résultats sont combinés pour déterminer la solution multiclasse finale. Dans ce cadre, deux stratégies principales ont été largement utilisées, à savoir la stratégie *one-vs-rest* et la stratégie *one-vs-one*. Selon (Hsu & Lin, 2002), les deux stratégies offrent souvent des performances similaires, tandis que des résultats contrastés sont observés par (Pawara *et al.*, 2020). Dans ce même contexte, on peut également mentionner la stratégie de "error-correcting output-coding" (Dietterich & Bakiri, 1994) et sa généralisation (Allwein *et al.*, 2000). La troisième approche consiste à construire une architecture d'apprentissage hiérarchique en supposant qu'il existe une certaine relation entre les étiquettes de classe. Ainsi, deux stratégies principales ont été proposées, à savoir l'approche basée sur les instances (Zupan *et al.*, 1999), et la stratégie basée sur les prédictions (Godbole *et al.*, 2002; Silva-Palacios *et al.*, 2017).

Dans cet article, nous proposons de tirer le meilleur parti des stratégies par décomposition et hiérarchique en intégrant les relations entre les étiquettes de classe dans une approche de type *one-vs-rest*. À cette fin, nous concevons une **stratégie multitâche**, dans laquelle chaque classifieur *one-vs-rest* est appris dans une configuration multitâche, ce qui permet de prendre en compte l'interdépendance des étiquettes de classe en une seule étape d'apprentissage, par opposition aux stratégies hiérarchiques qui nécessitent de deux étapes interdépendantes. Ainsi, chaque problème (ou tâche) *one-vs-rest* devrait bénéficier de l'apprentissage simultané des autres tâches comme le suggère (Caruana, 1998), si celles-ci sont corrélées ou montrent des similarités cognitives. En particulier, nous proposons deux architectures multitâches de type *one-vs-rest* (**tout-partagé** et **partagé-privé**) basées sur des algorithmes multitâches bien connus (Liu *et al.*, 2017). Ces deux architectures sont comparées à des solutions par décomposition et hiérarchiques de l'état de l'art en utilisant six jeux de données de référence pour l'analyse des sentiments (Socher *et al.*, 2013; Nakov *et al.*, 2016), la détection des émotions (Chatterjee *et al.*, 2019), la classification thématique (Greene & Cunningham, 2006) et la classification de relations lexico-sémantiques (Balikas *et al.*, 2019).

2 Stratégie *one-vs-rest* multitâche

La stratégie par décomposition *one-vs-rest* transforme un problème à N classes en N problèmes binaires (ici, tâches), où chaque classe doit être discriminée de toutes les autres classes. Ainsi, la décision finale est donnée par le classifieur avec la probabilité d'inférence maximale. Dans notre approche *one-vs-rest* multitâche, les N classifieurs binaires sont appris dans un cadre multitâche, ce qui permet d'obtenir une performance accrue pour chacune des tâches si celles-ci sont liées, i.e. s'il existe une relation (de similarité ou cognitive) entre les classes (Caruana, 1998). Nous proposons d'introduire deux modèles multitâches multiclasses : le modèle tout-partagé et le modèle partagé-privé. Dans l'architecture **tout-partagé** (figure 1), un réseau de neurones apprend une représentation partagée commune à toutes les tâches, à partir de laquelle toutes les décisions des classifieurs binaires doivent être apprises. Dans l'architecture **partagé-privé**, une couche privée est concaténée à la couche partagée, permettant à chaque classifieur binaire d'apprendre sa fonction de décision à partir d'informations spécifiques à la tâche mais aussi communes à toutes les tâches (Liu *et al.*, 2017).

Les deux modèles sont formellement définis comme suit pour exactement une couche générique partagée et une couche privée par classifieur binaire. Soit X_k un vecteur d'entrée [1], la couche partagée

1. $X_1 = X$, où X est le vecteur de texte encodé.

$S(X_k)$ est calculée comme dans l'équation 1, où W_{S^k} est une matrice de poids, b_{S^k} est un vecteur de biais [2], et $k \in [1, K]$ où K est le nombre de couches partagées.

$$S(X_k) = \sigma(W_{S^k} X_k + b_{S^k}) = X_{k+1} \tag{1}$$

La couche privée $H^j(Z_q)$ de chaque classifieur binaire indépendant j, qui résout la tâche T_j ($j \in [1, N]$) est définie dans l'équation 2, où $q \in [1, Q]$ et Q est le nombre de couches cachées. Notez que pour l'architecture tout-partagé, $Z_1 = S(X_K)$, et $Z_1 = S(X_K) \oplus X$ pour le modèle partagé-privé, où $\oplus$ représente la concaténation.

$$H^j(Z_q) = \sigma(W^j_{H^q} Z_q + b^j_{H^q}) = Z_{q+1} \tag{2}$$

La classe prédite est donnée par le maximum de toutes les probabilités prédites $O^1, ..., O^N$, qui sont définies dans l'équation 3.

$$O^j = \sigma(W^j_O H^j(Z_Q) + b^j_O) \tag{3}$$

Les paramètres sont mis à jour en minimisant l'entropie croisée binaire $E(O^j, y^j)$, où y^j est l'étiquette de référence. Ainsi, les poids de la couche partagée sont mis à jour en minimisant alternativement l'entropie croisée binaire de chaque classifieur, tandis que les couches privées ne sont mises à jour que pour une tâche donnée.

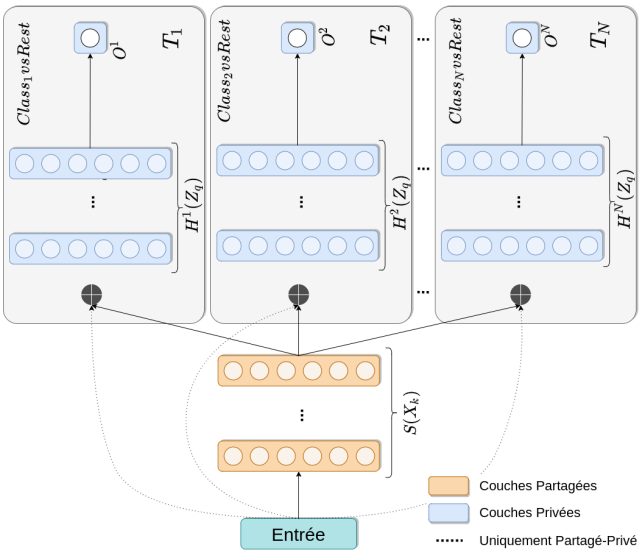


FIGURE 1 – Architectures tout-partagé et privé-partagé de la stratégie *one-vs-rest* multitâche. Les connexions en pointillés ne sont présentes que dans l'architecture partagé-privé.

3 Jeux de données et configurations expérimentales

Notre évaluation porte sur six jeux de données qui s'articulent autour de quatre applications connues de classification textuelle : l'analyse de sentiments, la détection des émotions, la classification thématique, l'identification de relations lexico-sémantiques. Pour l'analyse des sentiments, nous utilisons le jeu de données *Rotten Tomatoes* [3] (Socher *et al.*, 2013), qui consiste en des critiques de films notées sur une échelle de cinq valeurs, et le jeu de données Tweet2016 [4] (Nakov *et al.*, 2016), qui

2. Nous gardons la même notation pour le reste de l'article.
3. https ://www.kaggle.com/c/sentiment-analysis-on-movie-reviews
4. http ://alt.qcri.org/semeval2016/task4/

consiste en des tweets portant un avis sur les principales compagnies aériennes américaines (notation sur une échelle de cinq valeurs). Pour la détection des émotions, nous utilisons le jeu de données EmoContext [5] (Chatterjee *et al.*, 2019), qui consiste en des dialogues agent/utilisateur classés suivant quatre émotions (*happy*, *sad*, *angry*, *others*). Pour la classification thématique, nous utilisons les jeux de données BBC et BBC Sport [6] (Greene & Cunningham, 2006), qui consistent en des articles de presse classés suivant cinq domaines d'actualité, soit généraux pour BBC soit sportifs pour BBC Sport. Pour l'identification des relations lexico-sémantiques, nous créons un jeu de données spécifique RRW en concaténant trois jeux existants : RUMEN [7] (Balikas *et al.*, 2019), ROOT9 [8] (Santus *et al.*, 2016) and WEEDS [9] (Weeds *et al.*, 2014). RRW conjugue ainsi les relations d'hyperonymie, de co-hyponymie, de méronymie et de synonymie. Le détail est présenté dans le Tableau 1.

Jeu de données	Distribution des classes				
Rotten Tomatoes	*negative* 313	*somewhat negative* 686	*neutral* 508	*somewhat positive* 718	*positive* 402
Tweet2016	*very negative* 138	*negative* 2201	*neutral* 10081	*positive* 7830	*very positive* 382
BBC	*business* 510	*entertainment* 386	*politics* 417	*sport* 511	*tech* 401
BBC Sport	*athletics* 101	*cricket* 124	*football* 265	*rugby* 147	*tennis* 100
EmoContext	*happy* 4243	*sad* 5463	*angry* 5506	*others* 14948	- -
RRW	*co-hyponymy* 5283	*hypernymy* 12004	*meronymy* 2943	*random* 25741	*synonymy* 6728

TABLE 1 – Distribution des classes par jeu de données.

En ce qui concerne les configurations expérimentales, chaque jeu de données est aléatoirement divisé en trois sous-ensembles, à savoir l'ensemble d'entraînement (50 %), de validation (20 %) et de test (30 %). Afin de prendre en compte le déséquilibre des classes lors de la phase d'apprentissage [10], nous utilisons Adasyn (He *et al.*, 2008) pour équilibrer les instances sur les classes [11]. Au vu du Tableau 1, seul le jeu de données Tweet2016 reçoit ce pré-processus. Afin de coder l'entrée textuelle, nous utilisons l'architecture d'encodage pré-entraînée *Universal Sentence Encoder* [12] (Cer *et al.*, 2018) avec une dimension de 512 pour l'espace de représentation. Pour les relations lexico-sémantiques, chaque paire de mots est encodée par la concaténation des représentations GloVe (Pennington *et al.*, 2014) avec une dimension de 300.

Trois architectures sont implémentées comme bases de référence, une pour chaque paradigme multi-classe : stratégie native (*MultiClass*), par décomposition (*OneVsRest*), et hiérarchique (*Hierarchical*). *MultiClass* est un réseau de neurones de type *feed-forward* avec des couches cachées activées par une fonction sigmoïde, et la couche de sortie par une fonction *softmax*. L'entropie croisée binaire est utilisée comme fonction de perte. *OneVsRest* implémente une série de N réseaux de neurones de type *feed-forward* avec des couches cachées indépendantes. Toutes les couches sont activées par une fonction sigmoïde, l'entropie croisée binaire est utilisée comme fonction de perte, et le processus

5. https ://www.humanizing-ai.com/emocontext.html
6. https ://www.kaggle.com/c/learn-ai-bbc
7. https ://github.com/Houssam93/MultiTask-Learning-NLP/tree/master
8. https ://github.com/esantus/ROOT9
9. https ://github.com/SussexCompSem/learninghypernyms
10. Un problème bien connu des problèmes multiclasses.
11. Les ensembles de validation et de tests restent déséquilibrés.
12. https ://tfhub.dev/google/universal-sentence-encoder/4

de décision se base sur la fonction maximum. *Hierarchical* correspond à l'algorithme proposé par (Silva-Palacios *et al.*, 2017), que nous avons implémenté spécifiquement pour notre recherche.

Nos deux architectures *one-vs-rest* multitâches (OneVsRest Tout-Partagé et OneVsRest Partagé-Privé) reçoivent exactement la même configuration que le *OneVsRest* pour permettre une comparaison équitable. Le nombre de couches cachées ($K \in [1, 2]$ et $Q \in [1, 2]$), d'époques ($[1..100]$) et de neurones ($[5, 20, 50, 100, 150, 200, 300]$) sont des hyperparamètres optimisés par recherche par grille. Les poids sont initialisés avec une distribution uniforme échelonnée (Glorot & Bengio, 2010) et mis à jour à l'aide de Adam (Kingma & Ba, 2014) avec un taux d'apprentissage fixé à 0,001 [13].

4 Résultats

Chaque architecture est évaluée sur 25 exécutions différentes (partitions d'entraînement et de validation différentes mais avec l'ensemble de test identique) et la signification statistique est calculée pour six métriques différentes afin de refléter au mieux les différences entre les architectures. En particulier, nous calculons la F1 Micro, la F1 Macro, la F1 Pondérée, l'ACC Macro (précision), l'AUNU (surface moyenne sous la courbe) et l'AUNP (surface pondérée sous la courbe) (Hand & Till, 2001; Hossin & Sulaiman, 2015). Les résultats sont présentés dans le Tableau 2.

Les résultats montrent clairement que la stratégie*one-vs-rest* multitâche dépasse l'approche classique *one-vs-rest* en tenant compte de la relation entre les classes. Ceci est particulièrement pertinent pour Tweet2016, EmoContext et RRW, où les architectures tout-partagé et partagé-privé dépassent statistiquement les résultats de la stratégie *OneVsRest*. En particulier, pour Tweet2016, des améliorations maximales de 2,7% F1 Micro, 2,4% F1 Macro et 2,9% F1 pondérée peuvent être atteintes. Les améliorations pour RRW sont du même ordre, alors qu'elles sont moins expressives pour EmoContext, mais néanmoins statistiquement pertinentes. Pour *Rotten Tomatoes* et la classification thématique (c'est-à-dire BBC et BBC Sport), la situation diffère légèrement car seule une des architectures multitâches *one-vs-rest* dépasse statistiquement le *OneVsRest* : OneVsRest Tout-Partagé pour *Rotten Tomatoes* et BBC, et OneVsRest Partagé-Privé pour BBC Sport. En particulier, pour BBC Sport, des améliorations maximales de 0,7% F1 Micro, 0,7% F1 Macro et 0,7% F1 pondérée peuvent être atteintes. La différence entre les architectures tout-partagé et partagé-privé peut être due à la force de la relation entre les classes comme expliqué dans (Qureshi *et al.*, 2020). En effet, les architectures tout-partagé semblent être plus performantes que les architectures partagé-privé lorsque les classes sont fortement liées, alors que le contraire se produit lorsque les classes sont moins fortement liées.

La comparaison entre la stratégie multitâche de type *one-vs-rest* et les implémentations *MultiClass* et *Hierarchical* nécessite une analyse plus approfondie. Pour Tweet2016, EmoContext et RRW, la stratégie tout-partagé dépasse statistiquement les deux approches. En particulier, pour Tweet2016, des améliorations maximales de 0,8% (resp. 3%) F1 Micro, 0,8% (resp. 1,9%) F1 Macro et 0,9% (resp. 3,6%) F1 pondérée peuvent être atteintes par rapport à *MultiClass* (resp. *Hierarchical*). Ces résultats sont confirmés pour l'architecture partagé-privé dans le contexte de EmoContext et de RRW, mais pas pour Tweet2016, où les résultats ne peuvent pas être distingués de *MultiClass*. Pour BBC Sport, l'architecture partagé-privé est statistiquement plus performante que les résultats de *MultiClass*, mais les résultats sont similaires à ceux de *Hierarchical*. Il convient de noter que pour ce jeu de données, l'architecture *MultiClass* affiche les pires résultats, ce qui indique clairement la présence d'une relation entre les classes. Pour BBC, les résultats des stratégies multitâches *one-vs-rest* ne sont pas statistiquement supérieurs à ceux des stratégies *MultiClass* et *Hierarchical*, ce qui montre que ce jeu de données est certainement celui qui met en évidence le moins de relations entre les classes. Enfin,

13. Tous les codes sources sont disponibles sur demande.

	Algorithme	F1 Micro	F1 Macro	ACC Macro	AUNU	AUNP	F1 Pondérée
Tweet2016	*MultiClass*	0.514	0.362	0.805	0.663	0.655	0.537
	OneVsRest	0.495	0.346	0.798	0.653	0.641	0.517
	Hierarchical	0.492	0.351	0.797	0.656	0.634	0.510
	OneVsRest Tout-Partagé	**0.522**⋆+	**0.370**⋆†+	**0.809**⋆+	**0.666**⋆†+	**0.661**⋆†+	**0.546**⋆†+
	OneVsRest Partagé-Privé	0.514⋆+	0.362⋆+	0.806⋆+	0.662⋆+	0.653⋆+	0.539⋆+
Rotten Toma.	*MultiClass*	**0.403**	0.339	**0.761**	**0.598**	**0.605**	0.369
	OneVsRest	0.386	0.348	0.754	0.595	0.598	0.367
	Hierarchical	0.388	0.280	0.755	0.580	0.592	0.326
	OneVsRest Tout-Partagé	0.400⋆+	0.338+	0.760⋆+	0.596+	0.603⋆+	0.368+
	OneVsRest Partagé-Privé	0.393	**0.353**†+	0.757	0.600⋆+	0.602⋆+	**0.372+**
BBC	*MultiClass*	0.954	0.951	0.982	0.971	0.971	0.952
	OneVsRest	0.968	0.967	0.987	**0.980**	0.980	0.968
	Hierarchical	**0.969**	0.968	0.987	**0.980**	0.980	**0.969**
	OneVsRest Tout-Partagé	**0.969**⋆	**0.969**⋆	**0.988**⋆	**0.980**⋆	**0.981**⋆	**0.969**⋆
	OneVsRest Partagé-Privé	**0.969**	0.968	**0.988**	**0.980**	**0.981**	**0.969**⋆
BBC Sport	*MultiClass*	0.947	0.941	0.979	0.963	0.965	0.946
	OneVsRest	0.964	0.966	0.986	0.978	0.976	0.964
	Hierarchical	0.970	**0.974**	**0.988**	**0.983**	**0.980**	0.970
	OneVsRest Tout-Partagé	0.951	0.952	0.981	0.970	0.968	0.951
	OneVsRest Partagé-Privé	**0.971**⋆†	0.973⋆†	**0.988**⋆†	**0.983**⋆†	**0.980**⋆†	**0.971** ⋆†
EmoContext	*MultiClass*	0.825	0.808	0.912	0.869	0.866	0.825
	OneVsRest	0.824	0.806	0.912	0.866	0.864	0.823
	Hierarchical	0.821	0.804	0.911	0.868	0.864	0.820
	OneVsRest Tout-Partagé	**0.828**⋆†+	**0.810**⋆†+	**0.914**⋆†+	**0.872**⋆+	**0.869**⋆†+	**0.827**⋆†+
	OneVsRest Partagé-Privé	0.826⋆†+	0.809⋆+	0.913⋆†+	0.870	0.867⋆+	0.826⋆†+
RRW	*MultiClass*	0.587	0.494	0.835	0.686	0.704	0.583
	OneVsRest	0.597	0.51	0.839	0.687	0.702	0.59
	Hierarchical	0.608	0.512	0.843	0.688	0.708	0.59
	OneVsRest Tout-Partagé	0.617⋆†+	0.519†+	**0.847**⋆†+	0.697⋆†+	0.716⋆†+	0.602⋆†+
	OneVsRest Partagé-Privé	**0.618**⋆†+	**0.521**⋆†+	**0.847**⋆†+	**0.698**⋆†+	**0.717**⋆†+	**0.609**⋆†+

TABLE 2 – Moyenne des scores F1 Micro, F1 Macro, ACC Macro, AUNU, AUNP et F1 Pondérée sur 25 exécutions pour tous les jeux de données. La pertinence statistique est basée sur le test t en supposant des variances d'échantillon inégales, et ⋆, † et + mettent en evidence une valeur de $p \leq 0.05$ par rapport aux algorithmes de références *OneVsRest*, *Multiclass* et *Hierarchical* respectivement.

pour *Rotten Tomatoes*, notre modèle partagé-privé démontre des résultats statistiquement supérieurs par rapport à *MultiClass* pour la F1 Macro, où une amélioration de 1,4% peut être atteinte. Mais c'est la seule mesure où cela se produit. En ce qui concerne les autres métriques, nous obtenons des résultats similaires à ceux du *MultiClass* pour les deux versions *one-vs-rest* multitâches. Cependant, des améliorations significatives sont obtenues par les deux architectures multitâches par rapport à *Hierarchical*, à une exception près, avec des améliorations maximales de 1,2% de F1 Micro, 7,3% de F1 Macro et 4,6% de F1 pondérée.

Afin de mieux comprendre le comportements des différentes architectures, nous présentons leurs performances (F1 Micro) à nombre de paramètres identiques pour tous les jeux de données dans la figure 2. Les résultats montrent que l'architecture hiérarchique est celle qui donne de moins bons résultats quelques soit le niveau de paramètres. Inversement, l'architecture *one-vs-rest* partagé-privé met en évidence les résultats les plus élevés et stables indépendemment du nombre de paramètres, en particulier par rapport à la version multiclasse standard (*MultiClass*), surtout pour un niveau de paramètre faible. Ces résultats confirment les hypothèses des réseaux multitâches qui agissent comme des auto-régulateurs et permettent une meilleure généralisation (Caruana, 1998).

5 Conclusions

Dans cet article, nous définissons deux architectures multitâches de type *one-vs-rest* qui combinent des ensembles de classifieurs appris de manière multitâche pour tirer partie de la relation entre classes. En particulier, nous mettons en œuvre deux solutions différentes (tout-partagé et partagé-privé), qui diffèrent légèrement dans leurs hypothèses d'apprentissage. Les résultats obtenus sur six jeux de données de référence s'attaquant à quatre tâches différentes montrent que la stratégie multitâche de type *one-vs-rest* est toujours plus performante que l'algorithme classique de type *one-vs-rest*. L'approche multitâche est également statistiquement plus performante que les réseaux de neurones classiques et les algorithmes hiérarchiques, lorsque (1) les classes sont fortement liées et (2) les jeux de données contiennent un grand nombre d'exemples d'entraînement (voir tableau 1). Dans tous les autres cas, des résultats solides sont obtenus concurrençant clairement les stratégies de référence.

Il est important de noter que ces derniers algorithmes donnent des résultats avec une variance élevée selon le jeu de données en question, alors que notre approche est plus robuste à la variation (effet de bord de l'approche multitâche). Ainsi, nous pensons que notre approche peut se combiner favorablement à des modèles d'apprentissage spécifiquement adaptés à la tâche ou dédiés au texte. En effet, nous proposons une architecture multiclasse générique adaptable à tout problème de classification à plusieurs étiquettes de classe. Des études dans ce sens sont du ressort du travail futur, comme l'implémentation de nouvelles stratégies de décomposition, notamment impliquant la définition de multiples couches partagées pour des tuples de classes. Nous pensons également nous baser sur les résultats récents de (Akhmouch *et al.*, 2021) pour intégrer des caractéristiques dans le processus de décision et en étudier la distribution afin de maximiser une classification multitâche multiclasse.

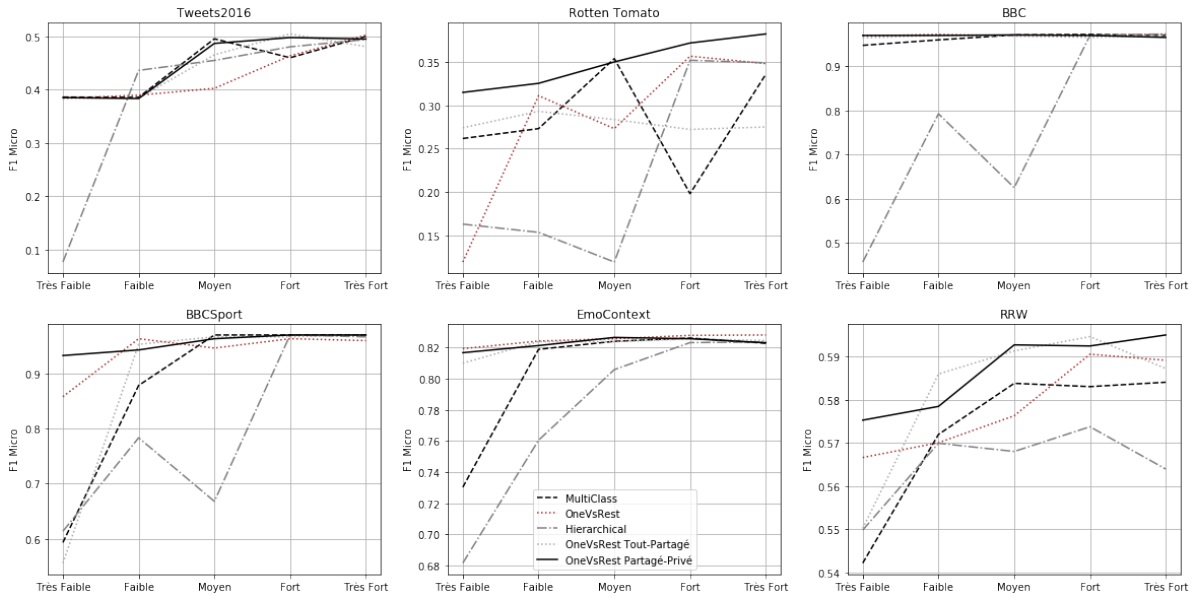


FIGURE 2 – Évolution du F1 Micro pour toutes les architectures avec des paramètres fixes pour tous les jeux de données. Afin de faciliter la lecture et du fait de l'impossibilité de garantir exactement le même nombre de paramètres, 5 niveaux ont été définis pour regrouper la taille des architectures explorées : *Très faible* entre 10k et 30k paramètres, *Faible* entre 30k et 100k, *Moyen* entre 100k et 200k, *Fort* entre 200k et 500k, et finalement, *Très Fort* entre 500k et 1M.

Références

AKHMOUCH H., DIAS G. & MORENO J. G. (2021). Understanding feature focus in multitask settings for lexico-semantic relation identification. In *Proceedings of the 59st Annual Meeting of the Association for Computational Linguistics*.

ALLWEIN E., SCHAPIRE R. & SINGER Y. (2000). Reducing multiclass to binary : A unifying approach for margin classifiers. *Journal of Machine Learning Research*, **1**, 113–141.

BALIKAS G., DIAS G., MORALIYSKI R., AKHMOUCH H. & AMINI M.-R. (2019). Learning lexical-semantic relations using intuitive cognitive links. In *41st European Conference on Information Retrieval (ECIR)*, p. 3–18.

BALIKAS G., MOURA S. & AMINI M.-R. (2017). Multitask learning for fine-grained twitter sentiment analysis. In *40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, p. 1005–1008.

CARUANA R. (1998). Multitask learning. In *Learning to learn*, p. 95–133. Springer.

CER D., YANG Y., KONG S., HUA N., LIMTIACO N., JOHN R. S., CONSTANT N., GUAJARDO-CESPEDES M., YUAN S., TAR C., SUNG Y., STROPE B. & KURZWEIL R. (2018). Universal sentence encoder. *CoRR*, **abs/1803.11175**.

CHATTERJEE A., NARAHARI K. N., JOSHI M. & AGRAWAL P. (2019). SemEval-2019 task 3 : EmoContext contextual emotion detection in text. In *13th International Workshop on Semantic Evaluation (SemEval)*, p. 39–48.

CONNEAU A., SCHWENK H., BARRAULT L. & LECUN Y. (2017). Very deep convolutional networks for text classification. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, p. 1107–1116.

CRAMMER K. & SINGER Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, **2**, 265–292.

DIETTERICH T. & BAKIRI G. (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**, 263–286.

GLOROT X. & BENGIO Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, p. 249–256.

GODBOLE S., SARAWAGI S. & CHAKRABARTI S. (2002). Scaling multi-class support vector machines using inter-class confusion. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, p. 513–518.

GREENE D. & CUNNINGHAM P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. In *23rd International Conference on Machine Learning (ICML)*, p. 377–384.

HAND D. & TILL R. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, **45**(2), 171–186.

HE H., YANG B., EDWARDO G. & SHUTAO L. (2008). Adasyn : Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, p. 1322–1328.

HOSSIN M. & SULAIMAN M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, **5**(2), 1–11.

HSU C.-W. & LIN C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, **13**(2), 415–425.

KINGMA D. & BA J. (2014). Adam : A method for stochastic optimization. *2nd International Conference on Learning Representations (ICLR)*.

LIU P., QIU X. & HUANG X. (2017). Adversarial multi-task learning for text classification. In *55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

NAKOV P., RITTER A., ROSENTHAL S., SEBASTIANI F. & STOYANOV V. (2016). SemEval-2016 task 4 : Sentiment analysis in Twitter. In *10th International Workshop on Semantic Evaluation (SemEval)*, p. 1–18.

OU G. & MURPHEY Y. L. (2007). Multi-class pattern classification using neural networks. *Pattern Recognition*, **40**(1), 4–18.

PAWARA P., OKAFOR E., GROEFSEMA M., HE S., SCHOMAKER L. & WIERING M. (2020). One-vs-one classification for deep neural networks. *Pattern Recognition*, **108**, 107528.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, p. 1532–1543.

QURESHI S., DIAS G., SAHA S. & HASANUZZAMAN M. (2020). Improving depression level estimation by concurrently learning emotion intensity. *IEEE Computational Intelligence Magazine*, **15**, 47–59.

SANTUS E., LENCI A., CHIU T.-S., LU Q. & HUANG C.-R. (2016). Nine features in a random forest to learn taxonomical semantic relations. In *10th International Conference on Language Resources and Evaluation (LREC)*, p. 4557–4564.

SILVA-PALACIOS D., FERRI C. & RAMÍREZ-QUINTANA M. J. (2017). Improving performance of multiclass classification by inducing class hierarchies. *Procedia Computer Science*, **108**, 1692–1701.

SOCHER R., PERELYGIN A., WU J., CHUANG J., MANNING C., NG A. & POTTS C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1631–1642.

TENG Z., VO D.-T. & ZHANG Y. (2016). Context-sensitive lexicon features for neural sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1629–1638.

WANG G., LI C., WANG W., ZHANG Y., SHEN D., ZHANG X., HENAO R. & CARIN L. (2018). Joint embedding of words and labels for text classification. In *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 2321–2331.

WEEDS J., CLARKE D., REFFIN J., WEIR D. & KELLER B. (2014). Learning to distinguish hypernyms and co-hyponyms. In *25th International Conference on Computational Linguistics (COLING)*, p. 2249–2259.

YAO L., MAO C. & LUO Y. (2019). Graph convolutional networks for text classification. In *33rd Conference on Artificial Intelligence (AAAI)*, p. 7370–7377.

YE Z., GENG Y., CHEN J., CHEN J., XU X., ZHENG S., WANG F., ZHANG J. & CHEN H. (2020). Zero-shot text classification via reinforced self-training. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 3014–3024.

ZHANG H. & ZHANG J. (2020). Text graph transformer for document classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 8322–8327.

ZHOU S. & LI X. (2020). Feature engineering vs. deep learning for paper section identification : Toward applications in chinese medical literature. *Information Processing & Management*, **57**(3), 102206.

ZUPAN B., BOHANEC M., DEMŠAR J. & BRATKO I. (1999). Learning by discovering concept hierarchies. *Artificial Intelligence*, **109**(1-2), 211–242.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

TREMoLo : un corpus multi-étiquettes de tweets en français pour la caractérisation des registres de langue

Jade Mekki[1, 2] Delphine Battistelli[2] Nicolas Béchet[3] Gwénolé Lecorvé[1]
(1) IRISA, 263 Avenue Général Leclerc, 35 000 Rennes, France
(2) Modyco, 200 Avenue de la République, 92 001 Nanterre, France
(3) IRISA, Campus de Tohannic – Rue Yves Mainguys, 56 000 Vannes, France
prenom.nom@irisa.fr, prenom.nom@parisnanterre.fr

RÉSUMÉ

Des registres tels que familier, courant et soutenu sont un phénomène immédiatement perceptible par tout locuteur d'une langue. Ils restent encore peu étudiés en traitement des langues (TAL), en particulier en dehors de l'anglais. Cet article présente un large corpus de tweets en français annotés en registres de langue. L'annotation intègre des marqueurs propres à ce type de textes (tels que les émoticônes ou les hashtags) et habituellement évincés dans les travaux en TAL. À partir d'une graine annotée manuellement en proportion d'appartenance aux registres, un classifieur de type CamemBERT est appris et appliqué sur un large ensemble de tweets. Le corpus annoté en résultant compte 228 505 tweets pour un total de 6 millions de mots. Des premières analyses statistiques sont menées et permettent de conclure à la qualité du corpus présenté. Le corpus ainsi que son guide d'annotation sont mis à la disposition de la communauté scientifique.

ABSTRACT

TREMoLo : a Multi-Label Corpus of French Tweets for Language Register Characterization

The casual, neutral, and formal language registers are a highly perceptible characteristic of discourse productions. However, they are still poorly studied in natural language processing, especially outside English, and for new textual types like tweets. To stimulate this line of research, this paper introduces a large corpus of French tweets annotated in language registers. It has been built on a preliminary detailed linguistic analysis of tweets. After training a multi-label CamemBERT classifier on a manually annotated subset, the whole corpus of tweets has been automatically labeled. The final corpus counts 228 505 tweets for a total of 6M words. Initial statistical analyses are conducted and allow us to conclude that the corpus presented is of good quality. The corpus and its annotation guide are available to the scientific community.

MOTS-CLÉS : registres de langue, CamemBERT, corpus annoté, tweets.

KEYWORDS: language registers, CamemBERT, annotated corpus, tweets.

1 Introduction

Le registre de langue dans lequel se situe un texte (à l'oral comme à l'écrit) apparaît comme un trait saillant. Il renvoie au contexte d'énonciation dans lequel il est — ou a été — produit (et qui comprend notamment la relation du locuteur avec ses interlocuteurs). Parmi les manifestations possibles de ce phénomène sociolinguistique, le partitionnement en registres tels que familier, courant et soutenu est

probablement le plus répandu. Si des corpus comme GYAFC (Rao & Tetreault, 2018) — où ce type de variations est appelé « niveau de formalité » — ont récemment popularisé le domaine, celui-ci est encore globalement peu étudié en traitement automatique des langues (TAL), et particulièrement en dehors de l'anglais. Par ailleurs, bien que de nouveaux types de textes aient émergé depuis les deux dernières décennies — tels que les tweets, et plus généralement ceux que l'on range sous le terme *Communications Médiées par Ordinateurs (CMO)* —, les travaux sur les registres de langue traitent surtout des types plus classiques de textes dont les caractéristiques sont plus ou moins connues de la littérature linguistique (on associera ainsi généralement par exemple les insultes au registre familier et la diversité de connecteurs logiques à du registre soutenu). Dès lors, les analyses de corpus CMO en termes de registres de langue constituent un défi tant pour la linguistique descriptive que pour les différentes applications en TAL. Pour répondre à ces enjeux, cet article présente le corpus TREMoLo [1], constitué de 228 505 tweets en français (6M mots), annotés en registres de langue familier, courant et soutenu. À partir d'une graine annotée manuellement, les annotations ont été généralisées à l'ensemble du corpus en utilisant un classifieur fondé sur CamemBERT. Après un état de l'art en section 2, la composition du corpus est présentée en section 3, les protocoles d'annotation manuelle et automatique sont décrits en sections 4 et 5. Enfin, la qualité du corpus et quelques premiers résultats statistiques sur la caractérisation des registres sont exposés en section 5.

2 État de l'art

En sociolinguistique, la notion de registre de langue fait globalement référence à la perception de variétés linguistiques associées à des situations de communication particulières (Todorov, 2013) et il est admis qu'un registre de langue peut être caractérisé par des motifs spécifiques (Ferguson, 1982). L'utilisation des termes *« niveau »*, *« style »* ou *« genre »* coexistent (Gadet, 1996; Bourquin, 1965; Joos, 1967) pour désigner ce phénomène, mais le terme *« registre »* semble tout de même tendre à prévaloir, du moins dans la littérature anglo-saxonne (Biber, 1991; Sanders, 1993; Ure, 1982). En linguistique de corpus, c'est ce dernier qui est retenu en particulier dans les travaux de (Biber & Conrad, 2019; Biber, 1991). Ils étudient quantitativement la présence de traits linguistiques définis *a priori* sur un corpus [2] selon différents axes : oral/écrit, formel/informel, etc. L'objectif est d'identifier les cooccurrences de traits selon ces axes. En TAL, pour l'anglais, (Peterson *et al.*, 2011; Pavlick & Tetreault, 2016) proposent des techniques de classification de textes en formel vs. informel à partir d'un corpus de courriers électroniques tandis que (Sheikha & Inkpen, 2010) utilise une régression pour prédire un niveau de formalité à partir d'un corpus de textes formel [3]/informel [4]. Pour le français, dans (Lecorvé *et al.*, 2019), les auteurs étudient conjointement une tâche de classification et de construction d'un corpus de données web [5] annoté en utilisant une approche semi-supervisée. Ces différents travaux présentent plusieurs limites : la composition des corpus montre différents biais en mélangeant les types de textes, les annotations manuelles ne suivent pas de guide d'annotation, et aucune des techniques de prédiction ne prend en compte le fait qu'un même texte puisse être perçu comme appartenant simultanément à plusieurs registres. De ce fait, la qualité de ces corpus annotés peut être discutée. L'approche que nous présentons dans cet article répond à ce besoin en proposant à la communauté un large corpus de données annotées en registres de langue pour le

1. https://gitlab.inria.fr/jmekki/tremolo_corpus
2. Les corpus *T2K-SWAL*, *LSWE* et *ARCHER*.
3. Les *Late Modern English Corpus*, *Enron Email Corpus* et *Open American National Corpus*.
4. Les *Reuters Corpus* et *Open American National Corpus*.
5. 400 000 pages web ont été collectées à partir de requêtes composées de lexiques familier, courant et soutenu.

français dont la qualité est assurée par : la suppression des biais liés à la présence de multiples types de textes, l'élaboration d'un guide d'annotation [6] pour l'annotation manuelle, et l'annotation automatique multi-étiquettes plus mimétique de la réalité.

3 Constitution du corpus

La constitution d'un corpus de textes écrits représentatif de l'usage réel des registres de langue présente deux difficultés majeures : tout d'abord le lien bi-univoque fort entre certains registres et certains types de textes (par exemple le soutenu associé à des romans de la littérature classique, le familier aux forums de discussion, et le courant à des dépêches journalistiques) ; ensuite l'association quasi immédiate de la modalité orale avec le registre familier d'une part, et de la modalité écrite avec les registres courant ou soutenu d'autre part (Gadet, 2000; Rebourcet, 2008). Pour répondre à ces biais, nous avons choisi de construire notre corpus à partir d'un seul type de textes issu des CMO définis comme « *toute communication humaine qui se produit à travers l'utilisation de deux ou plusieurs dispositifs électroniques* » (McQuail, 2010). Un des intérêts des CMO sur le plan linguistique réside dans le fait qu'ils contribuent à créer un « *parlécrit* » (Jacques, 1999) par le caractère instantané des échanges qu'ils matérialisent ; l'intérêt des tweets en particulier parmi les CMO est leur limite à 280 caractères, imposée par Twitter, ce qui homogénéise la taille des textes produits et analysés. L'extraction automatique des tweets a été conduite en s'appuyant sur l'hypothèse qu'en collectant les tweets qui contiennent les hashtags les plus utilisés à un moment donné (*TT* pour « Trending Topics ») dans une zone géographique donnée, la diversité des productions devrait être représentative des différentes fonctions du langage et de différents registres de langue. Aussi, si (Sinclair, 2005) avance que la constitution d'un corpus doit se fonder sur des « *critères externes* », c'est à dire les fonctions de communication des textes, cet article tend justement à en découvrir de nouvelles parmi les CMO : afin de ne pas poser d'*a priori* sur ces dernières elles n'ont pas été utilisées pour la constitution du corpus qui s'est basée sur les TT. L'API de Twitter [7] permet, à partir d'un identifiant de lieu (dans notre cas celui de Paris), de récupérer automatiquement les 50 TT. Pour chaque TT, une extraction a recherché tous les tweets le mentionnant. Afin de couvrir le plus grand nombre d'usages et donc de sujets différents, 10 extractions ont été faites à 10 dates différentes. Elles couvrent une durée totale d'un mois. Les tweets non français ont été repérés grâce à un module python [8] qui, pour un texte donné, prédit une langue à une certaine probabilité P. Si $P \leq 0.90$ pour le français, alors le texte est conservé dans le corpus. La valeur de P est fixée afin de garder des textes avec la présence de quelques termes non français intéressants tels que « *lol* », « *dead* », « *stan* »... Quant aux tweets tronqués, ils ont été repérés grâce à un signe de ponctuation spécifique de Twitter : trois points de suspension resserrés différents des « ... » classiques. Une règle symbolique a écarté les tweets qui se terminaient par ce signe de ponctuation particulier. Finalement, après l'exclusion des tweets non français ou tronqués, le corpus compte 228 505 tweets (6 201 339 mots).

6. https://hal.archives-ouvertes.fr/hal-03218217
7. https://developer.twitter.com/en/docs
8. https://pypi.org/project/langdetect/

4 Annotation manuelle

Une analyse linguistique du corpus est faite afin de proposer un guide d'annotation qui inclut certains éléments linguistiques spécifiques aux tweets. Une de nos contributions est de les intégrer au lieu de les écarter comme dans (Agarwal *et al.*, 2011; Pak & Paroubek, 2010; Go *et al.*, 2009).

Descripteurs linguistiques pour l'analyse des CMO (Paveau, 2013) utilise le terme « *technomorphèmes* » pour désigner les formes qui découlent des discours numériques. Parmi elles, les hashtags qui sont définis comme un ou plusieurs mots contigus précédés d'un # (par exemple « #Rentrée2020 »). Certaines typologies de hashtags mettent l'accent sur leur fonction d'indexation (Jackiewicz & Vidak, 2014). En plus d'intégrer à notre analyse ce type de fonction pour les hashtags car nous pensons qu'elle joue un rôle dans la perception de registres dans les tweets, nous proposons d'intégrer en outre le degré d'intégration syntaxique plus ou moins fort des hashtags. Un autre type de *technomorphèmes* est le pictogramme qui se réfère à la fois à un « émoticône » [9] et à un « emoji » [10]. Nous utilisons les trois fonctions de la typologie proposée par (Magué *et al.*, 2020) en les adaptant à l'analyse de notre corpus : la fonction de remplacement (quand un pictogramme remplace un syntagme) ; la fonction d'illustration (quand il a une fonction référentielle) ; la fonction de modalisation (quand il indique l'émotion ou l'attitude énonciative de l'auteur). Nous ajoutons une autre fonction : la fonction d'encadrement/structuration (lorsqu'il entoure ou pointe vers une information). En mettant à jour une liste issue d'une étude qui avait déjà identifié des descripteurs dans la littérature scientifique pour les registres de langue (Mekki *et al.*, 2018) avec ces traits spécifiques aux technomorphèmes, notre annotation prend au final en compte un ensemble de 52 descripteurs experts (dont certains sont présentés tables 1 et 2).

Protocole d'annotation Sur l'ensemble du corpus, 4 000 tweets (ou textes) ont été sélectionnés au hasard pour être annotés manuellement en proportion de registres de langue. Puisque nous divisons l'espace linguistique en 3 registres (familier, courant et soutenu), les catégories utilisées pour l'annotation sont les mêmes. Une catégorie « poubelle » est ajoutée pour les tweets mal encodés ou incompréhensibles. L'annotateur doit ordonner les registres en fonction de leur prédominance dans un texte en leur attribuant un rang [11] qui doit être justifié par la présence d'au moins un descripteur de la liste issue de l'analyse linguistique préliminaire. Chaque rang est ensuite transformé en proportion de registre. Soit :

- R un ensemble de registres ayant obtenu un rang et $Card(R)$ son nombre d'éléments,
- $r_i \in R$ un registre de R ayant obtenu le rang i,
- $inv(i)$ le rang inversé du rang i défini par $inv(i) = Card(R) - i + 1$
- srg la somme des rangs définie par $srg = \sum_{i=1}^{Card(R)} i$

La proportion du registre r_i est alors définie par $Prop_{r_i} = \frac{inv(i)}{srg}$

Ainsi, pour un texte annoté comme ceci, r_1=familier, r_2=soutenu et r_3=courant, on obtient :

- $R = \{r_1, r_2, r_3\}$ et $Card(R) = 3$,
- $inv(1) = 3$, $inv(2) = 2$, $inv(3) = 1$
- $srg = 6$

9. Un émoticône est un signe graphique ressemblant à une émotion (Beccucci, 2018).
10. Un emoji est un symbole répertorié dans une base de données (ibid.).
11. À noter que ne pas mettre de rang signifie que le registre n'est pas présent dans le texte selon l'annotateur.

Familier	TC	**F vs. Autres**	**Exemple**
Remplacement du « il » par « y »*	33,2	6,6% / 0,4%	*Y* sont pas sérieux
Répétitions de caractères*	29,5	11,3% / 0,5%	L*ooo*l
Onomatopées	22,5	11,7% / 0,7%	oh
Courant	TC	**C vs. Autres**	**Exemple**
Présent comme unique temps*	3,2	13,8% / 3,1%	on lui *coupe* le pied, il *doit* jouer
Agglutination en nom propre*	2,8	2,9% / 0,7%	*#ThierryBodson* revient sur
# sans relation syntaxique*	2,7	7,5% / 2,0%	*#fastfashion #slowfashion*
Soutenu	TC	**S vs. Autres**	**Exemple**
Inversion sujet verbe	12,8	20,0% / 1,5%	*As tu* lu X
Diversité des connecteurs logiques	7,4	20,0% / 2,6%	*car* [...] *et*
Discours rapporté*	6,8	38,2% / 5,3%	Merci chère amie, *dit-elle*

TABLE 1 – Top 3 des descripteurs (* : issus de notre analyse linguistique pour les CMO) qui caractérisent les registres dans le corpus annoté manuellement. Chaque descripteur est donné avec son taux de croissance et fréquences relatives associées, ainsi qu'un exemple. Le taux de croissance, noté *TC*, est introduit par l'équation 1.

Les proportions en registre sur cet exemple sont donc familier 50% ($\frac{3}{6}$), soutenu 33% ($\frac{2}{6}$) et courant 17% ($\frac{1}{6}$).

Chaque texte est annoté par 2 annotateurs experts [12]. Ne sont conservées que les étiquettes présentes dans l'intersection des 2 annotations. Pour chaque étiquette sa moyenne est calculée. Sur les 4 000 tweets annotés manuellement par 4 annotateurs, 976 textes ne sont pas dans l'intersection entre les 2 annotations. Une seconde annotation est alors faite par un 5ème annotateur expert. Après cette seconde annotation, 976 tweets sont dans l'intersection. Au final, 3 269 tweets annotés manuellement sont conservés, soit 81,73% des textes totaux de la graine.

Résultats de l'annotation manuelle Les résultats de l'annotation manuelle sont dominés par le registre courant (50,5% de la graine), puis par le familier (38,7%) et enfin le soutenu (10,2%). Afin de caractériser un registre de langue R_1 par rapport aux autres registres (notés R_2), l'importance de chaque descripteur D vu dans R_1 est mesurée en calculant son taux de croissance (TC) calculé à partir de sa fréquence relative, notée $Freq$, dans R_1 et R_2 :

$$TC(D_{R_1|R_2}) = \begin{cases} \infty, & si Freq_{R_2}(D) = 0 \\ \frac{Freq_{R_1}(D)}{Freq_{R_2}(D)}, & sinon \end{cases} \quad (1)$$

Si $TC(D_{R_1|R_2}) > 1$, alors D est émergent. Tous les TC du registre courant sont inférieurs à ceux du registre familier et soutenu, ces valeurs soulignent ses limites floues avec les autres registres (table 1). Au contraire, le familier présente des TC aux valeurs élevées qui indiquent la présence de formes très spécifiques pour ce registre. De plus, la présence de *technomorphèmes* pour les registres courant et soutenu signifie que certains éléments spécifiques aux CMO ont été intégrés à la norme grammaticale.

5 Annotation automatique

Notre objectif est d'obtenir un corpus entièrement annoté en registres, or nous ne disposons que de 3 269 textes annotés (graine). Notre approche consiste alors à augmenter l'ensemble de données

12. Les annotateurs sont des doctorant.e.s ou chercheur.e.s en sciences du langage spécialisé.e.s en écrits numériques ou en TAL.

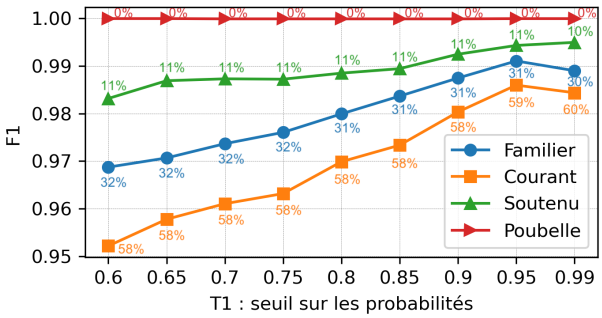


FIGURE 1 – $F1$ pour chaque registre et leur % dans l'ensemble des textes à ajouter à chaque valeur de $T1$

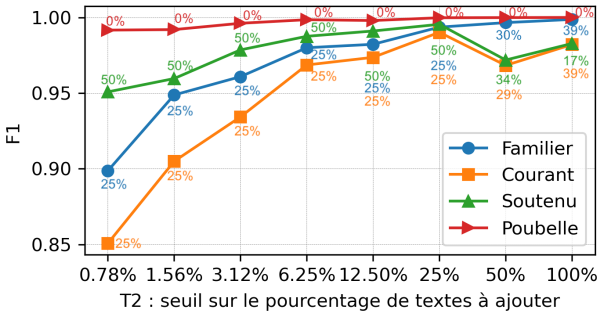


FIGURE 2 – $F1$ pour chaque registre et leur % de textes avec une probabilité ≥ 0.90 pour 1 des 4 registres à chaque valeur de $T2$

d'entraînement afin d'améliorer la qualité de prédiction finales des textes restants.

Méthodologie Tout d'abord, une phase de discrétisation des proportions est appliquée à chaque annotation A : $A \geq 50\% = 1$, et $A < 50\% = 0$. Puis, un modèle en deux étapes est adopté : l'apprentissage du classifieur multi-étiquettes est initialisé à partir de la graine. Un seuil est appliqué pour sélectionner les textes dont la prédiction est jugée fiable. Deux seuils indépendants sont introduits : soit sur la probabilité d'appartenance à un registre (noté $T1$), soit sur le nombre maximum de textes à ajouter (noté $T2$). Pour $T1$, tous les textes ayant une probabilité qui dépasse le seuil sont ajoutés. $T1$ garde la même valeur pour les quatre registres de langue. Pour $T2$, toutes les probabilités sont triées par ordre décroissant et pour chacun des 4 registres les $\frac{T2}{4}$ premiers textes sont ajoutés. Après avoir filtré les textes, le classifieur commence un second apprentissage à partir de la graine augmentée. Ses prédictions sont les prédictions finales. La *F-mesure* (notée $F1$) est utilisée pour évaluer les performances du modèle.

Expériences Pour les expériences, le modèle « base » de CamemBERT est utilisé comme classifieur (Martin *et al.*, 2020). Les paramètres sont fixés à 10^{-4} pour le taux d'apprentissage, 8 pour le nombre d'epochs, et une division de la graine 90%/10% d'entraînement/test. La figure 1 indique une détérioration des résultats entre 0,95 et 0,99 : le déséquilibre de la répartition des registres dans les données d'entraînement s'accentue légèrement à 0,99 (60% de textes courant, 30% familier et 10% soutenu). De manière générale, le classifieur est sûr de lui : plus de la moitié du corpus est ajouté à la graine (76%) avec le seuil le plus strict : $T1 = 0,99$. Une échelle logarithmique est prise pour faire varier $T2$. Les meilleurs scores sont obtenus lorsque $T2$ est fixé à 25% (figure 2). La légère dégradation des $F1$ à partir de 25% peut être due à la baisse du pourcentage de textes ayant une probabilité ≥ 0.90 pour 1 des 4 registres : il est de 100% lorsque $T2$ est à 25% et décroît à 93% et 95% lorsque $T2$ passe à 50% puis 100%. $T2$ à 25% (66 369 textes) semble donc être un bon équilibre entre la quantité et la qualité des données.

Analyse des résultats La distribution des registres est relativement similaire à celle de la graine annotée manuellement : 30,58% familier, 58,81% courant et 10,61% soutenu. Ces résultats suivent la tendance des annotations manuelles et indiquent la qualité des prédictions finales. De plus, pour les registres courant et soutenu, plusieurs traits émergents sont des *technomorphèmes* (table 2). Le registre courant montre un usage commercial des tweets qui met à profit la fonction d'indexation

Familier	TC	**F vs. Autres**	**Exemple**
Orthographe électronique	8.3	6.7% / 0.8%	Ha *ptdrrrr*
Remplacement du « il » par « y »	2.5	6.5% / 2.2%	*Y*'en a le 25
Motif « juste »*	2.1	0.5% / 0.2%	*Juste* comme ça
Courant	TC	**C vs. Autres**	**Exemple**
Absence d'un item attendu	2.2	0.1% / 0.07%	ils ⊘ vont quand même pas
# sans relation syntaxique*	1.6	11.4% / 7.3%	*#stress #bonheur*
# indépendant syntaxiquement*	1.4	12.5% / 8.7%	[...] . *#MondayMotivation*
Soutenu	TC	**S vs. Autres**	**Exemple**
Fonction d'encadrement	6.7	2.2% / 0.3%	🌲 [ECOLOGIE] 🌳 À #Montréal,
ou de structuration du pictogramme*			🔴 #X banni de #Facebook
			📣 [Webinar] J-1 « Le bilan à 6 ans »
Phrase sans ponctuation*	2.3	57.1% / 24.3%	VIDEO. Crise des transports *:*
# intégré syntaxiquement*	2.3	10.8% / 4.7%	les *#ViolencesPolicieres* ne sont pas

TABLE 2 – Top 3 des descripteurs (* : issus de notre analyse linguistique pour les CMO) qui caractérisent les registres dans le corpus annoté automatiquement. Chaque descripteur est donné avec son taux de croissance et fréquences relatives associées, ainsi qu'un exemple.

des hashtags afin de les rendre investigables, par exemple : « Le jeu *#MonstrumGame* de X sort ». Le phénomène d'agglutination pour des noms propres sert également à créer de nouveaux mots qui réfèrent à de nouveaux produits : « *#PokemonGO*, une nouvelle application ». Le registre soutenu quant à lui, intègre syntaxiquement les hashtags : « Les violences vécues en *#France* ne sont pas des *#incivilites* » ; et les pictogrammes comme du lexique : « les habitudes de #consommation en 🇫🇷 ». Enfin, le registre familier est utilisé pour dialoguer entre utilisateurs avec des marqueurs de l'oral : « Et *bim* dans tes dents », « *Eh* X il est timide *ou quoi* ? ». Ainsi, ces premières analyses confirment que les *technomorphèmes* ont été intégrés à la norme grammaticale et qu'ils ne sont plus uniquement caractéristiques du registre familier. Ils peuvent au contraire marquer un discours soutenu.

6 Conclusion

Dans cet article, nous avons présenté le corpus TREMoLo qui rassemble 228 505 tweets annotés en registres familier, courant et soutenu. Pour cela, une graine a été annotée manuellement en multiples étiquettes, en suivant un guide d'annotation issu d'une analyse linguistique du corpus. En utilisant le modèle CamemBERT, un enrichissement des données d'apprentissage a ensuite été réalisé afin finalement de prédire des registres pour l'ensemble du corpus. Nos expérimentations ont montré l'excellente qualité du corpus proposé. Suffisamment volumineux pour ouvrir la voie à de futurs travaux statistiques, il permettra de découvrir de nouvelles connaissances fondamentales sur les registres de langue.

Remerciements

Ce travail a bénéficié du soutien du projet TREMoLo [13] (ANR-16-CE23-0019) de l'Agence Nationale de la Recherche (ANR).

13. https://tremolo.irisa.fr/

Références

AGARWAL A., XIE B., VOVSHA I., RAMBOW O. & PASSONNEAU R. J. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, p. 30–38.

BECCUCCI L. (2018). Pierre halté, les émoticônes et les interjections dans le tchat. limoges : Éditions lambert lucas, 2018. *Communication et organisation*, (54), 253–255.

BIBER D. (1991). *Variation across speech and writing*. Cambridge University Press.

BIBER D. & CONRAD S. (2019). *Register, genre, and style*. Cambridge University Press.

BOURQUIN G. (1965). Niveaux, aspects et registres de langage. remarques à propos de quelques problèmes théoriques et pédagogiques. *Linguistics*, **3**(13), 5–15.

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

GADET F. (1996). Niveaux de langue et variation intrinsèque. *Palimpsestes*, **10**.

GADET F. (2000). Français de référence et syntaxe. *Cahiers de l'Institut de Linguistique de Louvain*, **26**(1-4), 265–283.

GO A., BHAYANI R. & HUANG L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, **1**(12), 2009.

JACKIEWICZ A. & VIDAK M. (2014). Étude sur les mots-dièse. In *shs Web of Conferences*, volume 8, p. 2033–2050 : EDP Sciences.

JACQUES A. (1999). Internet, communication et langue française.

JOOS M. (1967). *The five clocks*, volume 58. New York : Harcourt, Brace & World.

LECORVÉ G., AYATS H., FOURNIER B., MEKKI J., CHEVELU J., BATTISTELLI D. & BÉCHET N. (2019). Towards the automatic processing of language registers : Semi-supervisedly built corpus and classifier for french.

MAGUÉ J.-P., ROSSI-GENSANE N. & HALTÉ P. (2020). De la segmentation dans les tweets : signes de ponctuation, connecteurs, émoticônes et émojis. *Corpus*, (20).

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

MCQUAIL D. (2010). *McQuail's mass communication theory*. Sage publications.

MEKKI J., BATTISTELLI D., LECORVÉ G. & BÉCHET N. (2018). Identification de descripteurs pour la caractérisation de registres. In *Proceedings of Rencontres Jeunes Chercheurs (RJC) of the CORIA-TALN conference*.

PAK A. & PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, p. 1320–1326.

PAVEAU M.-A. (2013). Genre de discours et technologie discursive. tweet, twittécriture et twittérature. *Pratiques. Linguistique, littérature, didactique*, (157-158), 7–30.

PAVLICK E. & TETREAULT J. (2016). An empirical analysis of formality in online communication. *Transactions of the Association of Computational Linguistics*, **4**(1).

PETERSON K., HOHENSEE M. & XIA F. (2011). Email formality in the workplace : A case study on the enron corpus. In *Proceedings of the Workshop on Languages in Social Media*.

RAO S. & TETREAULT J. (2018). Dear sir or madam, may i introduce the gyafc dataset : Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv :1803.06535*.

REBOURCET S. (2008). Le français standard et la norme : l'histoire d'une «nationalisme linguistique et littéraire» à la française. *Communication, lettres et sciences du langage*, **2**(1), 107–118.

SANDERS C. (1993). *French today : language in its social context*. Cambridge University Press.

SHEIKHA F. A. & INKPEN D. (2010). Automatic classification of documents by formality. In *IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*.

SINCLAIR J. (2005). Corpus and text-basic principles. *Developing linguistic corpora : A guide to good practice*, **92**, 1–16.

TODOROV T. (2013). *Mikhaïl Bakhtine. Le principe dialogique. Suivi de : Ecrits du Cercle de Bakhtine*. Le Seuil.

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

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Un modèle Transformer Génératif Pré-entraîné pour le _____ français

Antoine Simoulin[1, 2] Benoit Crabbé[2]
(1) Quantmetry, 8 rue d'Anjou, 75008 Paris, France
(2) Université de Paris, Olympe de Gouges, 8 Rue Albert Einstein, 75013 Paris
asimoulin@quantmetry.fr, benoit.crabbe@linguist.univ-paris-diderot.fr

RÉSUMÉ

Nous proposons une adaptation en français du fameux modèle *Generative Pre-trained Transformer* (GPT). Ce dernier appartient à la catégorie des architectures transformers qui ont significativement transformé les méthodes de traitement automatique du langage. Ces architectures sont en particulier pré-entraînées sur des tâches auto-supervisées et sont ainsi spécifiques pour une langue donnée. Si certaines sont disponibles en français, la plupart se déclinent avant tout en anglais. GPT est particulièrement efficace pour les tâches de génération de texte. Par ailleurs, il est possible de l'appliquer à de nombreux cas d'usages. Ses propriétés génératives singulières permettent de l'utiliser dans des conditions originales comme l'apprentissage sans exemple qui ne suppose aucune mise à jour des poids du modèle, ou modification de l'architecture.

ABSTRACT

Generative Pre-trained Transformer in _____ (French)

We introduce a French adaptation from the well-known GPT model. GPT relies on transformer pre-trained architectures, which profoundly transformed natural language processing methods. Such models are pre-trained using a self-supervised objective and are therefore specific to a given language. Although some models may exist in French, the majority is released in English only. GPT achieves impressive language generation performances. The model can address a large variety of tasks. In particular, the model may benefit from original configurations such as few-shot or zero-shot learning. In such configurations, it is possible to address tasks without any parameters fine-tuning.

MOTS-CLÉS : GPT, Génératif, Transformer, Pré-entraîné, français.

KEYWORDS: GPT, Transformer, Generative, Pre-trained, French.

1 Introduction

L'utilisation des modèles pré-entraînés et des architectures à base de transformers (Vaswani *et al.*, 2017) a significativement amélioré les performances des modèles de traitement automatique du langage (TAL). Les modèles GPT (Radford *et al.*, 2019a,b; Brown *et al.*, 2020) cherchent à améliorer le potentiel de génération automatique de texte en proposant un paradigme d'apprentissage et d'inférence qui permet d'utiliser le même modèle sur plusieurs tâches sans ajustement de son architecture. Cette particularité peut être étendue jusqu'à une configuration d'apprentissage sans exemple (en anglais, *zero-shot learning*). Dans cette configuration, les poids du modèle ne sont pas mis à jour sur des exemples spécifiques à une tâche.

Volume 1 : conférence principale, pages 246–255.

La généralité affichée par les modèles transformers a néanmoins ses limites. En premier lieu, le pré-entraînement est spécifique à la langue utilisée. L'adaptation de tels modèles dans d'autres langues est loin d'être trivial. Elle nécessite en particulier de très larges corpus — jusqu'à plusieurs milliards de *tokens*. L'entraînement des modèles requiert par ailleurs une importante puissance de calcul, cet étape est typiquement réalisée sur plusieurs dizaines de GPUs ou TPUs. Finalement, l'analyse de ces modèles et l'étude de leur pertinence suppose la disponibilité de *benchmarks* d'évaluation rigoureux.

A notre connaissance, aucun modèle transformer génératif n'a encore été proposé pour le français. Nous adaptons ainsi les modèles OpenAI GPT et GPT-2 (Radford *et al.*, 2019a,b). Nos contributions sont les suivantes :

— Nous proposons un corpus spécifique à l'entraînement des modèles de langues neuronaux en français. La construction du corpus est détaillée en SECTION 2.
— Nous avons entraîné deux modèles avec un très grand nombre de paramètres que nous diffusons en accès ouvert afin de favoriser leur étude et application dans un contexte académique comme industriel. L'architecture des modèles utilisés est décrite en SECTION 3.
— Nous proposons des *benchmarks* d'évaluations spécifiques aux modèles de langue, inspirés des *benchmarks* anglais pour favoriser la comparaison avec des modèles de langue alternatifs. L'ensemble des méthodes d'évaluation sont présentées en SECTION 4.

2 Construction des corpus

Constitution du corpus d'entraînement Les modèles transformers n'ont besoin que de texte brut pour le pré-entraînement. Néanmoins, le nombre important de paramètres nécessite l'utilisation de larges corpus. Dans le cas des modèles GPT, les documents utilisés sont plus longs que ceux utilisés pour des modèles tels que BERT (Devlin *et al.*, 2019). La plupart des corpus utilisés pour les modèles BERT en français : Camembert (Martin *et al.*, 2020) ou Flaubert (Le *et al.*, 2020a,b) s'appuient sur des documents relativement courts dont l'ordre n'était pas conservé. Nous avons donc dû préparer des corpus spécifiques. Nous avons agrégé deux corpus d'ordre de grandeurs différents pour entrainer les modèles. On résume leurs statistiques en TABLE 1.

Le premier corpus, utilisé pour l'entraînement de GPT$_{fr}$-124M, est une agrégation de corpus existants : Wikipédia, OpenSubtitle (Tiedemann, 2012) et Gutenberg. Les documents sont séparés en phrases. Les phrases successives sont ensuite concaténées dans la limite de 1 024 *tokens* par document.

Modèles	OpenAI GPT	OpenAI GPT-2	GPT$_{fr}$-124M	GPT$_{fr}$-1B
# Documents ($\times 10^6$)	2,26†	8,00	1,66	7,36
# *Tokens* ($\times 10^9$)	1,16†	4,68†	1,60	3,11
Moy. # *tokens* par document	512†	585†	965	422

TABLE 1 – Caractéristiques des corpus utilisés pour le pré-entraînement des modèles. Les données marquées † sont estimées à partir des caractéristiques disponibles. En particulier, pour le modèle OpenAI GPT on confond le nombre de *tokens* par document et la taille du contexte. Les caractéristiques du modèle OpenAI GPT-2 sont estimées à partir de l'échantillon proposé en accès ouvert.

Le second corpus est utilisé pour l'entraînement de notre modèle avec un milliard de paramètres : GPT$_{fr}$-1B. Il augmente le premier avec des données extraites du *Common Crawl* (Li *et al.*, 2019)

en français. Les données *Common Crawl* sont filtrées en plusieurs étapes en nous inspirant de la procédure proposée dans Brown *et al.* (2020). Dans un premier temps, nous avons exclu tous les documents courts, de moins de 128 *tokens* comme proposé dans Shoeybi *et al.* (2019). Ce filtre très simple a permis de filtrer plus de 93% des documents du corpus original. Nous avons ensuite filtré les documents dont la distribution des mots est très éloignée de celle de notre premier corpus. Pour cela nous avons entraîné un classifieur binaire à discriminer les documents issus de notre premier corpus et du *Common Crawl* en utilisant 200 000 documents tirés aléatoirement. Nous avons exclu tous les documents présentant une probabilité <10% d'être issu du corpus d'entraînement. Ce seuil volontairement assez large nous a permis d'écarter les documents explicitement mal formés ou incohérents. Nous avons finalement appliqué un second filtre selon la structure des documents. Nous avons sélectionné les documents présentant une perplexité [1] faible selon notre modèle GPT$_{fr}$-124M. Afin de conserver des documents hors de la distribution, on considère pour chacun un seuil g. Avec g une réalisation d'une loi de Pareto $G \sim \mathcal{G}(\alpha)$. Le document est conservé si sa perplexité ppl vérifie : $g > ppl/ppl_{seuil}$. Avec le seuil de perplexité ppl_{seuil} fixé à 60.

Corpus pour l'évaluation des modèles de langues Toujours dans une optique de s'aligner avec les travaux en anglais, nous avons constitué deux corpus d'évaluation d'un modèle de langue à partir de l'encyclopédie en ligne Wikipédia. Pour cela nous avons collecté le texte des articles en français, labélisés « articles de qualité » et « bons articles ». [2] Nous avons collecté le texte de 2 246 articles de qualité et 3 776 bons articles sur une période de 2003 à 2020. Nous n'avons pas appliqué de pré-traitements spécifiques. En effet, l'utilisation des modèles transformers suppose généralement d'utiliser une tokenization spécifique qui assure que très peu de *tokens* sont en dehors du vocabulaire. Les caractéristiques des corpus et de leurs homologues anglais sont présenté dans la TABLE 2. Nous précisons que **ces articles ont été spécifiquement exclus du corpus utilisé pour le pré-entraînement de nos modèles**.

Nous avons divisé aléatoirement les articles de qualité en des corpus d'entraînement/validation/test contenant respectivement 2 126/60/60 articles pour constituer le corpus **WikiText-2-FR**. Pour le corpus **WikiText-72-FR**, nous avons conservé les corpus de validation et de test inchangés. Le jeu d'entraînement correspond à la concaténation du jeu d'entraînement **WikiText-35-FR** et de l'ensemble des bons articles.

	WikiText-EN				WikiText-FR			
	Valid	Test	Train-2	Train-103	Valid	Test	Train-35	Train-72
Documents	60	60	600	28 475	60	60	2 126	5 902
Tokens ($\times 10^3$)	218	246	2 089	103 227	896	897	35 166	72 961
Vocabulaire			33 278	267 735			137 589	205 403
Hors du vocabulaire			2,6%	0,4%			0,8%	1,2%

TABLE 2 – Statistiques descriptives des corpus **WikiText-FR**. La taille du vocabulaire est évaluée en utilisant le tokenizer MOSES (Koehn *et al.*, 2007). La proportion de mots hors du vocabulaire correspond au nombre de *tokens* apparaissant moins de trois fois.

1. Etant donné une séquence $U = \{u_1 \cdots u_n\}$, on défini sa perplexité : $PPL(U) = exp\left(-\frac{1}{T}\sum_{i=1}^{T} logp_\theta(u_i|u_{<i})\right)$ avec $logp_\theta(u_i|u_{<i})$ la log-vraisemblance conditionnelle, selon notre modèle, du ith token sachant les *tokens* précédents $u_{<i}$.

2. L'extraction du texte d'articles Wikipédia n'est pas une opération triviale. Afin de nous assurer de la bonne qualité des documents, nous avons directement utilisé l'API Wikipédia pour extraire les articles pour ce corpus.

3 Modèles

Architecture du modèle Nous avons utilisé l'architecture proposée pour les modèles anglais OpenAI GPT [3]. Le modèle est pré-entraîné selon une tâche de modèle de langue. Étant donné un corpus d'entraînement décrit comme une séquence de *tokens* $U = \{u_1 \cdots u_n\}$, on optimise les paramètres Θ du modèle pour maximiser la log-probabilité suivante : $\mathcal{L}(U) = \sum_i logP\left(u_i|u_{i-k} \cdots u_{i-1}; \Theta\right)$. Avec k la taille de la fenêtre de contexte. L'architecture de GPT s'appuie sur des transformers et présente beaucoup de similarités avec BERT. Il s'agit de plusieurs couches successives de décodeurs telles que définies dans l'architecture transformers. À la différence de BERT, le modèle applique une attention multi-têtes uniquement sur les *tokens* qui précèdent la position considérée.

Configuration du modèle à l'inférence Une fois le modèle pré-entraîné, il est possible de l'utiliser comme les modèles transformers classiques. On ajoute ainsi une couche spécifique à la tâche en sortie du modèle. On ajuste ensuite l'ensemble des paramètres de manière incrémentale (en anglais, *fine-tuning*) en fonction des exemples $x_1 \cdots x_m$ et des labels correspondants y.

Il est également possible de formaliser les tâches pour tirer parti des propriétés génératives du modèle. On transforme alors le jeu de données en séquences $x_1 \cdots x_m[SEP]y$. Chaque tâche est ainsi formalisée comme un modèle de langue : le modèle doit "générer" le label y comme la suite de la séquence $x_1 \cdots x_m[SEP]$. Il n'est alors pas nécessaire de modifier l'architecture du modèle. Il est également possible de résoudre la tâche sans mise à jour des poids du modèle (apprentissage sans exemple). Cette configuration est illustrée pour le résumé automatique en SECTION 4.

Architectures Nous avons entraîné deux modèles, dont l'un avec plus d'un milliard de paramètres, détaillés en TABLE 3. En nous appuyant sur les travaux de Shoeybi *et al.* (2019), qui comparent de nombreuses configurations, nous avons proposé une architecture permettant de ne pas utiliser de parallélisation du modèle. En effet la répartition du modèle en module dispersés sur plusieurs unités de calcul est un facteur de ralentissement important pour l'entraînement. Finalement les modèles utilisent un vocabulaire de type bytepair encoding (BPE) avec 50 000 unités (Sennrich *et al.*, 2016) entraîné sur l'ensemble du corpus utilisé pour l'entraînement de GPT$_{fr}$-124M.

Infrastructures L'entraînement de GPT$_{fr}$-124M a été effectué sur un TPU v2-8 à partir de l'interface Google Colab. L'entraînement du modèle Fr GPT$_{fr}$-1B a été mené en utilisant le supercalculateur français Jean Zay. Un cumul de 140 heures de calcul a été effectué sur du matériel de type Tesla V100 (TDP de 300W). L'entraînement a été distribué sur 4 noeuds de calcul de 8 GPUs. Nous avons utilisé de la parallélisation de données afin de diviser chaque micro batch sur les unités de calcul. Les émissions totales sont estimées à 580.61 kgCO$_2$eq. [4]

Entraînement Nous avons gardé le même paramétrage pour les deux modèles. Le taux d'apprentissage est fixé à $1.5e^{-4}$ avec une phase d'échauffement (en anglais, *warm up*) de 2 000 itérations puis une décroissance (en anglais, *decay*) selon une fonction cosinus. Nous avons effectué 125 000

3. Nous nous sommes en particulier appuyés sur la librairie Transformers.
4. Les estimations ont été réalisées à l'aide du Machine Learning Impact calculator présenté dans (Lacoste *et al.*, 2019).

itérations en utilisant une taille de batch de 128 documents et la précision mixte (Micikevicius *et al.*, 2018). Les autres paramètres (initialisation, dropout ...) sont fixés selon Radford *et al.* (2019a).

Modèles	OpenAI GPT	OpenAI GPT-2	GPT$_{fr}$-124M	GPT$_{fr}$-1B
Taille du contexte	512	1 024	1 024	1 024
# Couches	12	48	12	24
# Têtes d'attentions	12	25	12	14
Dimension des *embeddings*	768	1 600	768	1 792
# Paramètres ($\times 10^6$)	117	1 558	124	1 017

TABLE 3 – Caractéristiques des architectures et comparaison avec les modèles OpenAI (Radford *et al.*, 2019a,b).

4 Evaluation

Modèle de langue Le premier intérêt du modèle est de générer du texte cohérent. Pour évaluer la perplexité de nos modèles, nous avons utilisé les corpus présentés en SECTION 2. Le corpus utilisé pour le pré-entraînement étant proche de celui utilisé pour l'évaluation, nous n'avons pas effectué d'entraînement supplémentaire. Nous présentons les résultats sur le jeu d'évaluation en TABLE 4. Nous précisons que nous évaluons la perplexité sur la base de la tokenization définie par le modèle. Cette dernière est identique pour les modèles GPT$_{fr}$-124M et 1B mais peut être différente pour d'autres modèles. En particulier, nous avons considéré un modèle de langue de type 5-grams avec lissage de kneser-ney (Ney *et al.*, 1994) en utilisant l'outil SRILM (Stolcke, 2002) comme référence.

Les approches ne sont pas directement comparables car la tokenization est différente et notre modèle est entraîné sur un volume de données beaucoup plus conséquent. Les résultats en TABLE 4 sont donc donnés à titre illustratif mais soulignent la performance de notre modèle GPT$_{fr}$-1B.

Modèles	Modèle 5-grams	GPT$_{fr}$-124M	GPT$_{fr}$-1B
WikiText-35-FR (ppl)	166,7	109,2	12,9
WikiText-72-FR (ppl)	99,1		

TABLE 4 – Perplexité de nos modèles. Nous n'avons pas mis à jour les modèles sur le jeu d'entraînement et la perplexité est directement mesurée sur le jeu de test qui sont identiques pour deux *benchmarks*. Le modèle n-gram est entraîné sur les corpus d'entraînements correspondants.

Résumé automatique Cette tâche permet d'exploiter les propriétés génératives du modèle. Nous utilisons la configuration proposée dans (Radford *et al.*, 2019a) qui permet d'utiliser le modèle sans ajuster son architecture. Nous rajoutons simplement le motif *"Pour résumer :"*[5] après le texte original pour encourager le modèle à générer un texte qui résume les articles. Les poids du modèle ne sont donc pas mis à jour et aucune donnée d'entraînement n'est utilisée.

5. Dans le modèle OpenAI GPT-2, le motif rajouté est "TL ;DR :" qui correspond à "Too Long ; Didn't Read." et qui est utilisé sur le forum Reddit comme marqueur pour résumer une discussion.

Nous avons considéré le jeu de données OrangeSum pour le résumé abstratif (Eddine *et al.*, 2020). Nous générons la suite du texte en utilisant la stratégie de top-k *random sampling* (Fan *et al.*, 2018) avec $k = 2$. On retient les 3 premières phrases parmi les 100 *tokens* générés. Nous comparons notre modèle à la référence qui consiste à considérer la première phrase du texte comme résumé. Nous comparons les métriques ROUGES [6] (Lin, 2004) en TABLE 5. Les métriques montrent que, dans cette configuration complexe, nos modèles parviennent tout juste à s'approcher de la référence proposée.

	Synthèse			Titre		
	R1	R2	RL	R1	R2	RL
Première phrase	22,1	7,1	15,3	18,6	7,7	15,0
GPT$_{fr}$-124M	17,5	3,1	12,1	13,9	2,3	9,7
GPT$_{fr}$-1B	16,6	3,4	11,5	10,2	2,6	8,4

TABLE 5 – Comparaison des résumés générés avec le titre de l'article ou la synthèse proposée. Nous utilisons le score ROUGE et le corpus OrangeSum (Eddine *et al.*, 2020). Nos modèles sont utilisés en apprentissage sans exemple et donc sans mise à jour des paramètres sur le jeu d'entraînement.

Nous avons analysé manuellement des exemples. Le texte généré est correct au niveau de l'ortographe et de la syntaxe. Il s'inscrit également dans le thème et dans la continuité des articles proposés. Néanmoins le texte généré se concentre généralement sur un détail spécifique de l'article pour venir ensuite l'étoffer en inventant parfois des éléments (Kryscinski *et al.*, 2019). La méthode, si elle permet de générer du texte cohérent, ne parvient pas à synthétiser complètement l'idée générale du texte.

***Benchmark* FLUE** Les modèles génératifs élargissent certaines perspectives des modèles de type BERT. Néanmoins, ce type de pré-entraînement ne permet pas d'atteindre les mêmes performances qu'avec un modèle prenant en compte l'ensemble du contexte. Lorsque que l'on compare directement les modèles anglais sur le *benchmark* GLUE, on observe une différence moyenne de plus de 4 points entre OpenAI GPT et BERT-base (Radford *et al.*, 2019a). Nous avons tout de même comparé notre modèle sur le *benchmark* FLUE en français en TABLE 6.

Modèles	Livres	CLS DVD	Musique	PAWS-X	XNLI	Moy.
mBERT†	86,2	86,9	86,7	89,3	76,9	85,2
CamemBERT†	92,3	93,0	94,9	90,1	81,2	90,3
FlauBERT-base†	93,1	92,5	94,1	89,5	80,6	90,0
FlauBERT-large†	95,0	94,1	95,9	89,3	83,4	91,5
GPT$_{fr}$-124M	88,3	86,9	89,3	83,3	75,6	84,7
GPT$_{fr}$-1B	91,6	91,4	92,6	86,3	77,9	88,0

TABLE 6 – Scores de précisions pour les tâches discriminatives du *benchmark* FLUE. Le symbole † désigne les scores rapportés de Le *et al.* (2020b,a).

6. Les métriques ROUGES sont une collection de métriques permettant de comparer des résumés automatiques avec un texte de référence en calculant la proportion de "n-grams" communs entre les deux textes. ROUGE-1 (R1) correpond aux uni-grams, ROUGE-2 (R2) aux bi-gram et ROUGE-L (RL) à la plus longue séquence commune de n-gram rapportée au nombre d'uni-gram de la référence.

Nous avons considéré les tâches suivantes :

— CLS est un jeu de données composé d'avis sur Amazon à classer comme positifs ou négatifs. Il contient 3 catégories de produits : livres, DVD et musique. Chaque cartégorie est divisée en 2 000 exemples d'entraînements, de validation et d'évaluation.

— PAWS-X contient des paires de phrases. Il s'agit d'une tâche de classification binaire pour identifier les paires dont les deux phrases sont sémantiquement équivalentes. Il y a 49 401 exemples pour l'entraînement, 1 992 pour la validation et 1 985 pour l'évaluation.

— XNLI contient des paires de phrases. La tâche consiste à prédire si la première (prémisse) implique la seconde (hypothèse). 392 702 paires sont utilisées pour l'entraînement, 2 490 pour la validation et 5 010 pour l'évaluation.

Cette fois-ci les poids de notre modèle sont mis à jour. Les hyper-paramètres sont fixés selon les recommandations de Le *et al.* (2020a,b). Comme attendu, les performances du modèle ne se hissent pas à celles obtenues avec des modèles de type BERT.

Utilisation sans apprentissage Le modèle GPT-3 (Brown *et al.*, 2020) peut être adapté pour de nombreux cas d'usages, simplement en décrivant la consigne de la tâche suivie par un certain nombre d'exemples (en anglais, *zero-shot* et *few-shots learning*). Cette méthode cherche à conditionner le comportement du modèle en formatant le texte proposé en entrée en fonction de la tâche à réaliser. Les résultats sont surprenants mais les mécanismes sous-jacents restent largement à explorer. Néanmoins, il semblerait que le nombre de paramètres soit l'un des facteurs clés pour le fonctionnement de cette méthode. Notre modèle semble ainsi moins performant que GPT-3 sur des questions de culture générale ou de logique. Par exemple, quand on soumet le texte suivant : "Si Jérôme est plus grand que Michel, qui est le plus petit ?" le modèle GPT$_{fr}$-1B génère "Michel" mais ce résultat nous a semblé difficile à reproduire pour des expériences analogues. Si l'on cherche à générer la suite de la phrase suivante, "Quatre plus quatre font" le modèle va générer "quatre" alors que GPT-3 obtient la bonne réponse pour des expériences similaires.

5 Conclusion et travaux futurs

Nous proposons une version française du modèle GPT. S'il n'égalise pas les performances brutes de BERT, ses propriétés génératives permettent de l'utiliser dans des configurations remarquablement flexibles. Comme illustré dans nos expérimentations pour le résumé automatique, l'utilisation d'une configuration sans apprentissage reste très difficile pour le modèle. Cette configuration ouvre néanmoins des perspectives différentes d'un apprentissage traditionnel. Finalement, nous espérons que les performances de modèles de langues obtenues favoriseront son utilisation pour des cas d'usages correspondants. À cet effet, l'ensemble de nos contributions, incluant les modèles et les données sont disponibles en accès ouvert.

Remerciements

Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2020-AD011011823 attribuée par GENCI.

Références

BROWN T. B., MANN B., RYDER N., SUBBIAH M., KAPLAN J., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., AGARWAL S., HERBERT-VOSS A., KRUEGER G., HENIGHAN T., CHILD R., RAMESH A., ZIEGLER D. M., WU J., WINTER C., HESSE C., CHEN M., SIGLER E., LITWIN M., GRAY S., CHESS B., CLARK J., BERNER C., MCCANDLISH S., RADFORD A., SUTSKEVER I. & AMODEI D. (2020). Language models are few-shot learners. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

DEVLIN J., CHANG M., LEE K. & TOUTANOVA K. (2019). BERT : pre-training of deep bidirectional transformers for language understanding. In J. BURSTEIN, C. DORAN & T. SOLORIO, Éds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, p. 4171–4186 : Association for Computational Linguistics. DOI : 10.18653/v1/n19-1423.

EDDINE M. K., TIXIER A. J. & VAZIRGIANNIS M. (2020). Barthez : a skilled pretrained french sequence-to-sequence model.

FAN A., LEWIS M. & DAUPHIN Y. N. (2018). Hierarchical neural story generation. In I. GUREVYCH & Y. MIYAO, Éds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1 : Long Papers*, p. 889–898 : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1082.

KOEHN P., HOANG H., BIRCH A., CALLISON-BURCH C., FEDERICO M., BERTOLDI N., COWAN B., SHEN W., MORAN C., ZENS R., DYER C., BOJAR O., CONSTANTIN A. & HERBST E. (2007). Moses : Open source toolkit for statistical machine translation. In J. A. CARROLL, A. VAN DEN BOSCH & A. ZAENEN, Éds., *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic* : The Association for Computational Linguistics.

KRYSCINSKI W., KESKAR N. S., MCCANN B., XIONG C. & SOCHER R. (2019). Neural text summarization : A critical evaluation. In K. INUI, J. JIANG, V. NG & X. WAN, Éds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, p. 540–551 : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1051.

LACOSTE A., LUCCIONI A., SCHMIDT V. & DANDRES T. (2019). Quantifying the carbon emissions of machine learning. *CoRR*, **abs/1910.09700**.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020a). Flaubert : des modèles de langue contextualisés pré-entraînés pour le français (flaubert : Unsupervised language model pre-training for french). In C. BENZITOUN, C. BRAUD, L. HUBER, D. LANGLOIS, S. OUNI, S. POGODALLA & S. SCHNEIDER, Éds., *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelle, Nancy, France, June 8-19, 2020*, p. 268–278 : ATALA et AFCP.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020b). Flaubert : Unsupervised language model pre-training for french. In N. CALZOLARI, F. BÉCHET, P. BLACHE, K. CHOUKRI, C. CIERI, T. DECLERCK, S. GOGGI, H. ISAHARA, B. MAEGAARD, J. MARIANI, H. MAZO, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, p. 2479–2490 : European Language Resources Association.

LI X., MICHEL P., ANASTASOPOULOS A., BELINKOV Y., DURRANI N., FIRAT O., KOEHN P., NEUBIG G., PINO J. & SAJJAD H. (2019). Findings of the first shared task on machine translation robustness. In O. BOJAR, R. CHATTERJEE, C. FEDERMANN, M. FISHEL, Y. GRAHAM, B. HADDOW, M. HUCK, A. JIMENO-YEPES, P. KOEHN, A. MARTINS, C. MONZ, M. NEGRI, A. NÉVÉOL, M. L. NEVES, M. POST, M. TURCHI & K. VERSPOOR, Éds., *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2 : Shared Task Papers, Day 1*, p. 91–102 : Association for Computational Linguistics. DOI : 10.18653/v1/w19-5303.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out*, p. 74–81.

MARTIN L., MÜLLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In D. JURAFSKY, J. CHAI, N. SCHLUTER & J. R. TETREAULT, Éds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, p. 7203–7219 : Association for Computational Linguistics.

MICIKEVICIUS P., NARANG S., ALBEN J., DIAMOS G. F., ELSEN E., GARCÍA D., GINSBURG B., HOUSTON M., KUCHAIEV O., VENKATESH G. & WU H. (2018). Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* : OpenReview.net.

NEY H., ESSEN U. & KNESER R. (1994). On structuring probabilistic dependences in stochastic language modelling. *Comput. Speech Lang.*, **8**(1), 1–38. DOI : 10.1006/csla.1994.1001.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2019a). Improving language understanding by generative pre-training. OpenAI Blog : Improving Language Understanding with Unsupervised Learning.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019b). Language models are unsupervised multitask learners. OpenAI Blog : Better Language Models and Their Implications.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1 : Long Papers* : The Association for Computer Linguistics. DOI : 10.18653/v1/p16-1162.

SHOEYBI M., PATWARY M., PURI R., LEGRESLEY P., CASPER J. & CATANZARO B. (2019). Megatron-lm : Training multi-billion parameter language models using model parallelism.

STOLCKE A. (2002). SRILM - an extensible language modeling toolkit. In J. H. L. HANSEN & B. L. PELLOM, Éds., *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002* : ISCA.

TIEDEMANN J. (2012). Parallel data, tools and interfaces in OPUS. In N. CALZOLARI, K. CHOUKRI, T. DECLERCK, M. U. DOGAN, B. MAEGAARD, J. MARIANI, J. ODIJK & S. PIPERIDIS,

Éds., *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, p. 2214–2218 : European Language Resources Association (ELRA).

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. VON LUXBURG, S. BENGIO, H. M. WALLACH, R. FERGUS, S. V. N. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, p. 5998–6008.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Une étude des avis en ligne: généralisabilité d'un modèle d'évaluation

Hyun Jung Kang Iris Eshkol-Taravella
MoDyCo UMR7114, 200 Avenue de la République, 92001 Nanterre, France
clinguist.hjkang@gmail.com, ieshkolt@parisnanterre.fr

RÉSUMÉ

Ce travail se situe dans la continuité de nos travaux antérieurs proposant le modèle d'évaluation portant sur des avis en ligne sur des restaurants. Le modèle est composé de quatre catégories : l'opinion (positive, négative, mixte), la suggestion, l'intention et la description. Cet article vise à tester la généralisabilité du modèle en l'appliquant sur deux corpus supplémentaires : un corpus relevant d'un autre domaine (celui de l'hôtellerie) et un corpus écrit dans une autre langue (le coréen). Nous avons présenté l'annotation manuelle et la détection automatique de ces catégories en nous appuyant sur différents modèles de l'apprentissage de surface (SVM) et l'apprentissage profond (LSTM).

ABSTRACT

A Study of Online Reviews : Generalizability of the Evaluation Model

This work is a continuation of our previous work proposing a model of evaluation on online restaurant reviews. The model comprises four categories : opinion (positive, negative, mixed), suggestion, intention, and description. This article attempts to test the generalizability of the proposed model on two other corpora : a corpus from another domain (hotel) and a corpus written in another language (Korean). We presented manual annotation and automatic detection of these categories based on traditional machine learning (SVM) and deep learning (LSTM).

MOTS-CLÉS : fouille d'opinion, avis en ligne, classification supervisée, généralisabilité, coréen.

KEYWORDS: opinion mining, online reviews, supervised learning, generalizability, Korean.

1 Introduction

Dans un monde interconnecté confronté à l'augmentation exponentielle du volume de données, de nombreux internautes se réfèrent au bouche-à-oreille électronique (eWOM) d'inconnus à travers diverses plateformes (blogs, forums ou sites dédiés à la critique) (Wachsmuth *et al.*, 2014). Ainsi, les évaluations diffusées en ligne ont un pouvoir conséquent car elles influencent la prise de décision des internautes. En traitement automatique des langues (TAL), elles sont concernées par la tâche de la fouille d'opinions. Cependant, à la différence de nombreux travaux dans ce domaine, nous dépasserons largement la notion d'opinion positive ou négative. Dans cette optique, nous nous interrogerons sur la manière dont une évaluation émerge dans le langage en tant que jugement axiologique. L'évaluation renvoie au second degré de la subjectivité (Kerbrat-Orecchioni, 1980), à laquelle l'axiologie positive ou négative du locuteur est associée. Elle fait intervenir un ensemble de normes et de valeurs qui s'inscrit dans le cadre des perceptions collectives, sensibles aux différences sociales et culturelles, et qui évolue au fil du temps.

Dans ce cadre, nous avons proposé un modèle d'évaluation fondé sur l'observation manuelle du corpus d'avis postés en ligne sur des restaurants (Eshkol-Taravella & Kang, 2019; Kang & Eshkol-Taravella, 2020), que nous appellerons « RestoFR [1] ». Le modèle décrit différents types d'évaluation, cette dernière se composant de quatre catégories : l'opinion (positive/négative/mixte), la suggestion, l'intention et la description. L'opinion représente l'évaluation de la valeur du restaurant dans une dimension axiologique qui comporte les polarités positive, négative et mixte (e.g., les adjectifs évaluatifs, les lexiques des émotions ou des sentiments et les modifieurs). La suggestion vise d'une part à améliorer les produits (ou services) du restaurant et d'autre part à donner des conseils aux autres consommateurs (e.g., les verbes de parole, le mode impératif et le mode conditionnel). L'intention renvoie au souhait du client de renouveler ou non son expérience dans le restaurant, permettant aux restaurateurs de se renseigner sur les actions futures qu'un client a l'intention d'engager (e.g., les verbes au futur et le préfixe verbal d'itération « re- »). La description concerne les informations factuelles associées à l'expérience vécue, qui révèle aux lecteurs l'arrière-plan de cette dernière. Nous avons présenté la détection automatique de ces catégories fondée sur différents modèles de l'apprentissage de surface (Naïve Bayes, SVM, Logistic Regression) et l'apprentissage profond (CNN, LSTM) (Eshkol-Taravella & Kang, 2019; Kang & Eshkol-Taravella, 2020). La meilleure F-mesure a été obtenue grâce au classifieur SVM, donnant un score de 0,88 [2].

L'une des étapes essentielles de l'élaboration d'un modèle consiste à évaluer sa généralisabilité (Clark & Watson, 2016). La possibilité de généraliser est essentielle dans le contexte de la mondialisation, et particulièrement de l'ère numérique, dans lequel nous ne pouvons plus nous limiter aux données textuelles d'un seul domaine ou d'une seule langue. Dans cet article, nous avons pour objectif de vérifier si le modèle élaboré pour le corpus RestoFR peut s'appliquer à un corpus relevant d'un autre domaine (celui de l'hôtellerie) et à un corpus écrit dans une autre langue (le coréen). Si notre approche est valable, les six catégories d'évaluation devraient également être identifiées au sein des nouveaux corpus. L'annotation manuelle et la détection automatique réalisées sur les avis concernant des hôtels (HotelFR) sont décrites dans la seconde section. En ce qui concerne le corpus coréen (RestoKR), présenté dans la troisième section, nous nous en sommes tenues à son annotation manuelle car le prétraitement de cette langue est complexe et diffère largement de celui du français.

2 Application du modèle au corpus portant sur l'hôtellerie

La réussite de la généralisation d'une approche dans un autre domaine dépend principalement de deux éléments (Szarvas *et al.*, 2012) : d'abord, les domaines source et cible doivent être étroitement liés pour permettre le partage des connaissances ; deuxièmement, l'adaptation doit se fondre sur les points communs des deux domaines, tout en garantissant les caractéristiques particulières du domaine cible. Bien que l'hôtellerie et la restauration utilisent des lexiques différents, il s'agit dans les deux domaines de situations vécues dans un lieu, dont l'évaluation porte sur le processus de satisfaction et la valorisation de l'expérience. Trois hôtels situés à Paris ont été choisis au hasard, et pour chacun de ceux-ci, nous avons extrait d'un site internet [3] 99 avis rédigés en français. Les avis ont été extraits selon leur date de parution : ceux qui sont antérieurs au mois de février 2020 ont été collectés, dans la

1. Il s'agit de 6 287 avis collectés sur un site internet (https://www.lafourchette.com), correspondant à 17 268 phrases, avec une moyenne de dix mots par phrase.

2. Le nombre limité de pages ne permet pas de décrire le modèle et les traits linguistiques de manière plus détaillés. Nous pouvons mettre à votre disposition les articles concernés.

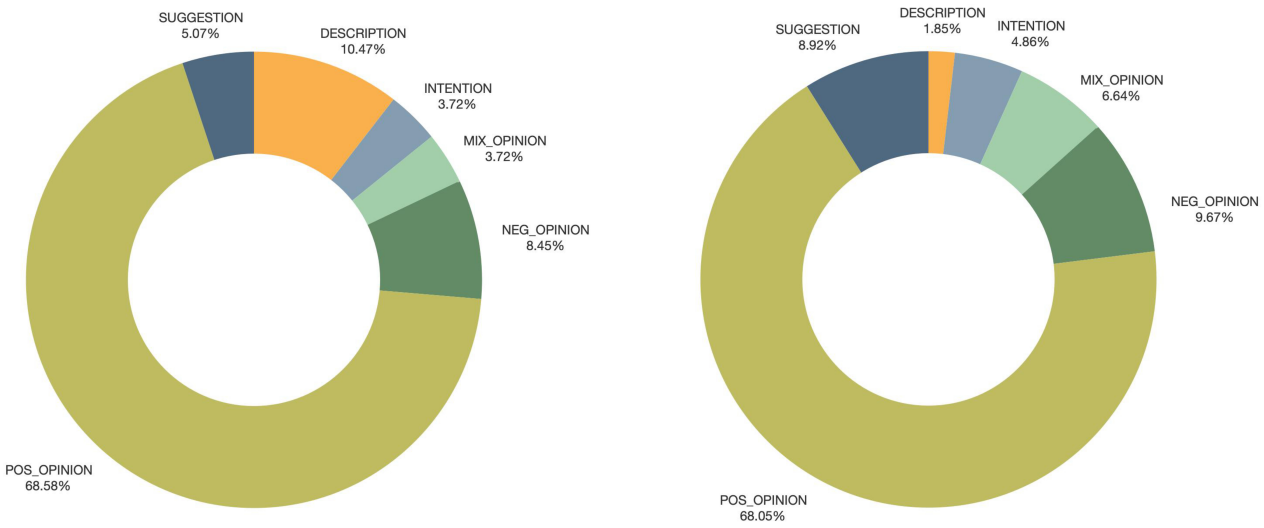
3. https://www.tripadvisor.com [consulté le 3er janvier 2021].

limite de 33 avis. Ils ont été segmentés en phrases en fonction des signes de ponctuation, produisant un total de 296 phrases (HotelFR). Le corpus a été annoté manuellement selon la typologie d'évaluation élaborée (POS_OPINION, NEG_OPINION, MIX_OPINION, SUGGESTION, INTENTION et DESCRIPTION), qui a ensuite été utilisée pour sa détection automatique.

2.1 Annotation manuelle

La tâche d'annotation a été réalisée par deux doctorantes en linguistique. Ces dernières ont suivi le guide d'annotation qui préconise d'attribuer une étiquette à chaque phrase. En cas d'ambiguïté, lorsque plusieurs catégories étaient pertinentes, les catégories les moins représentées dans le corpus (l'intention ou la suggestion) ont été retenues [4]. Pour valider la typologie d'évaluation et évaluer sa généralisabilité, l'accord inter-annotateur entre ces deux annotateurs est calculé en appliquant la mesure Kappa de Cohen (Cohen, 1960). Nous avons obtenu un accord inter-annotateur (AIA) de 0,83, considéré comme 'presque parfait' selon l'échelle de Landis & Koch (1977). Ce score montre que les catégories et les règles d'annotations sont bien définies et expliquées dans le guide d'annotation et que la typologie proposée peut être généralisée à d'autres domaines.

Néanmoins, l'observation du corpus a permis de révéler une différence significative par rapport à RestoFR. La répartition des catégories dans ce corpus, présentée dans la figure 1a, s'est avérée non homogène. En comparaison de RestoFR (voir figure 1b), la proportion de description s'est révélée être significativement plus importante, passant de 1,85 % à 10,47 %. Cette croissance est due aux particularités de l'hôtellerie, secteur dans lequel le confort est un aspect primordial ; les équipements, les installations et la localisation sont des éléments qui permettent de le garantir, qui font partie dans notre typologie d'une catégorie de description.



(a) HotelFR (b) RestoFR

FIGURE 1: Répartition des catégories

4. Ce choix a été fait dans la lignée de certains travaux de psychologie (Carreiras *et al.*, 1995; Gernsbacher, 1990; Kim *et al.*, 2004) affirmant que le contenu présenté au début de la phrase semble avoir un effet plus important dans la détermination du jugement final.

2.2 Annotation automatique

Nous avons considéré trois corpus d'entraînement différents : H (67 % de HotelFR de manière stratifiée), R (l'ensemble de RestoFR) et H+R (l'ensemble de RestoFR et 67 % de HotelFR de manière stratifiée). L'évaluation a été effectuée sur le corpus de test qui représente 33 % de HotelFR de manière stratifiée (98 phrases). En règle générale, nous avons adopté la méthodologie exploitée précédemment dans auteur1 & auteur2 (2019, 2020), dont nous ne présentons ici que la synthèse.

Nettoyage et prétraitement. La procédure comprend les étapes suivantes : le passage des mots en minuscules, le remplacement des chiffres par « NUM », le remplacement des émoticônes par « emoPOS » ou « emoNEG » selon leur polarité, la normalisation des abréviations (resto en restaurant, par exemple) et la lemmatisation au moyen de StanfordCoreNLP [5].

Classification. Parmi les divers algorithmes, nous avons choisi ceux ayant produit les meilleurs résultats sur le corpus RestoFR pour l'apprentissage de surface et l'apprentissage profond (SVM et LSTM respectivement). Le classifieur SVM a été appliqué à l'aide de la bibliothèque scikit-learn [6] (Pedregosa *et al.*, 2011) et le second avec les bibliothèques Keras (Chollet *et al.*, 2015) et TensorFlow. En ce qui concerne le classifieur SVM, les données textuelles ont d'abord été représentées selon la méthode de représentation vectorielle (`CountVectorizer` ou `TfidfVectorizer`) et à l'aide du paramètre n_gram. Leur meilleure combinaison a été obtenue grâce à une procédure de grille de recherche (`GridSearch`). Les caractéristiques prises en compte lors de l'apprentissage sont : les catégories morphosyntaxiques jugées pertinentes proposées par StanfordCoreNLP, les différentes variations des verbes, la négation, la conjonction mais, les mots positifs et négatifs, les scores de polarité et de subjectivité obtenus avec TextBlob [7], la position de la phrase dans l'avis, le nombre de caractères, la longueur de la phrase, la diversité et la densité lexicales, les ponctuations multiples et l'unité monétaire. Afin d'exploiter la technique des LSTM, nous avons pris en entrée une matrice d'embedding à l'aide de Word2vec. Cette dernière a été créée à nouveau pour chaque jeu d'entraînement. Par la suite, nous avons appliqué une couche avec 100 unités, envoyée à une couche dense avant une activation softmax. Les hyperparamètres choisis sont l'optimiseur Adam et une perte d'entropie, et la taille du batch était de cinq, avec sept époques. Pour toutes les expériences, une validation croisée stratifiée à cinq plis a été effectuée et le paramètre `class_weight` au mode équilibré (`'balanced'`) a été appliqué afin de résoudre le problème de la disproportion des classes.

Résultats. Pour chaque jeu de données d'entraînement, la moyenne pondérée de la précision, du rappel et de la F-mesure a été calculée lorsque SVM et LSTM ont été appliqués (tableau 1). Le classifieur SVM entraîné avec H+R donne le meilleur score, dont la F-mesure est de 0,8064. Grâce à la combinaison des données (HotelFR et RestoFR), les performances de SVM et LSTM se sont considérablement améliorées.

5. Stanford CoreNLP, https://stanfordnlp.github.io/CoreNLP/download.html [consulté le 3 janvier 2021].

6. http://scikit-learn.org/stable/ [consulté le 3 janvier 2021].

7. Une bibliothèque pour le traitement des données textuelles (https://textblob.readthedocs.io/en/dev/index.html).

TABLE 1: Moyenne pondérée de la F-mesure (HotelFR)

Jeu d'entraînement	**F-mesure**	
	SVM	**LSTM**
H	0,6366	0,0189
R	0,7992	0,5160
H+R	**0,8064**	**0,6684**

D'après une observation manuelle du corpus, il existe des moyens systématiques relativement indépendants du domaine pour exprimer une évaluation. Grâce à des constructions linguistiques communes, les données de RestoFR semblent avoir comblé le manque d'informations de HotelFR, malgré le fait que les deux ensembles de données proviennent de domaines différents (restaurants et hôtels). Ainsi, lorsque nous ne disposons pas de suffisamment de données annotées pour un domaine en particulier, celles relatives à un domaine qui s'en rapproche peuvent pallier ce déficit et donc permettre de réduire les coûts d'annotation nécessaires pour couvrir le manque des données.

TABLE 2: Précision, rappel et F-mesure de chaque catégorie (H+R et SVM)

	POS_OPINION	NEG_OPINION	MIX_OPINION	SUGGESTION	INTENTION	DESCRIPTION
Précision	0,84	0,67	0,67	1,00	1,00	0,60
Rappel	0,97	0,44	0,50	0,80	1,00	0,30
F-mesure	0,90	0,53	0,57	0,89	1,00	0,40

La précision et le rappel du meilleur résultat (H+R avec SVM) pour chaque catégorie sont présentés dans le tableau 2. L'intention est détectée avec la meilleure performance (la F-mesure étant 1,00), suivie par l'opinion positive et la suggestion dont la F-mesure se situe autour de 0,90. En revanche, la description possède la plus mauvaise performance (0,40). Ce résultat s'explique en partie par le manque d'échantillons des descriptions et donc par le faible nombre de caractéristiques fournies durant l'entraînement. De plus, cette catégorie est très hétérogène et varie en fonction du profil du client, ce qui cause l'apparition d'un large éventail de vocabulaires et de contextes. Ainsi, notre typologie d'évaluation peut être généralisée à d'autres domaines tant que l'évaluation porte sur des expériences à un endroit donné (un lieu touristique ou un magasin par exemple).

3 Application du modèle sur le corpus en langue coréenne

L'évaluation, qui émerge dans le langage en tant que jugement axiologique, peut être perçue et interprétée de différentes manières selon la culture et la langue. Plus précisément, l'interprétation axiologique de la valeur – ce qui est bon ou mauvais, agréable ou désagréable, satisfaisant ou insatisfaisant, souhaitable ou à éviter – peut varier d'une culture à l'autre. Cette section a donc pour but d'évaluer la généralisabilité de notre modèle d'évaluation proposé sur une autre langue, le coréen (RestoKR). Nous avons choisi six restaurants qui se trouvent à Séoul et proposent de la nourriture coréenne. Les avis sur les restaurants ont été sélectionnés en fonction de leur date de parution : pour

chaque restaurant, nous avons extrait dix des avis les plus récents du mois de mai 2020. Le corpus coréen est donc constitué de 60 avis (246 phrases), collectés sur le même site que le corpus HotelFR. Bien que les avis soient rédigés en coréen, ils sont comparables à ceux de RestoFR puisqu'il s'agit des évaluations effectuées par les clients. Les études coréennes sur la fouille d'opinions exploitent différentes méthodes de classification (l'apprentissage non supervisé, l'apprentissage de surface et l'apprentissage profond). Cependant, en raison de la spécificité du coréen en tant que langue agglutinante, de nombreuses études adoptent des méthodes particulières s'appuyant non sur des mots (comme pour le français) mais sur d'autres unités comme les morphèmes ou les jamos [8]. Ainsi, le prétraitement et la détection automatique sont complexes et bien différents de ceux du français. Dans cette étude, nous nous sommes donc limités à son annotation manuelle.

3.1 Annotation manuelle

Deux annotatrices coréennes (une doctorante en linguistique et une autre en littérature) ont effectué l'annotation manuelle du corpus RestoKR. La tâche d'annotation a consisté à attribuer à chaque phrase une étiquette parmi des catégories prédéfinies. Selon la mesure Kappa de Cohen (Cohen, 1960), nous avons obtenu un AIA de 0,92, un accord considéré comme « presque parfait » (Landis & Koch, 1977). Ce score a ainsi permis d'appuyer la reproductibilité de notre typologie d'évaluation sur une autre langue. La figure 2 illustre la répartition des catégories, qui a permis de mettre en lumière les spécificités du corpus coréen.

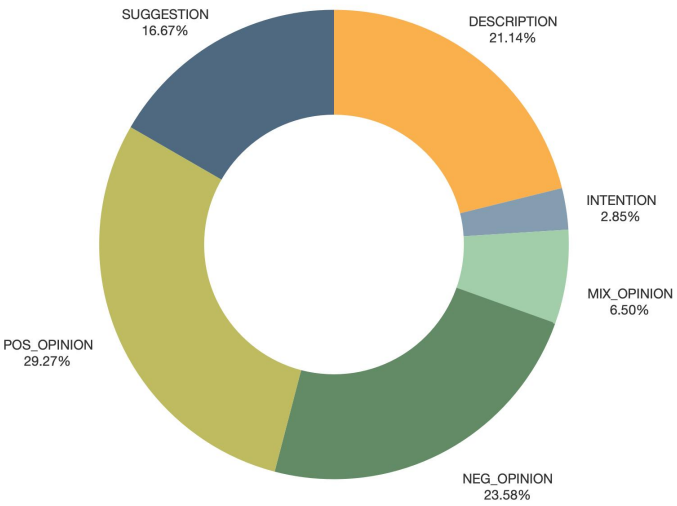


FIGURE 2: Répartition des catégories d'évaluation (RestoKR)

En premier lieu, la répartition des catégories est plus équilibrée que celle des corpus français (HotelFR, RestoFR) (cf. figures 1a, 1b). L'opinion positive constitue toujours la classe majoritaire (29,27 %), mais de façon moins marquée que dans RestoFR (68,05 %). À l'inverse, l'opinion négative atteint 23,58 %, contre 9,5 % pour le corpus RestoFR. Bien que Jurafsky *et al.* (2014) aient mis en évidence le principe de pollyanna – une tendance générale à préférer les informations positives – dans leur

8. Un ensemble d'unités phonétiques de base représentant les consonnes et les voyelles de la langue coréenne.

corpus anglais, il semble que celui-ci ne s'applique pas à notre corpus coréen [9].

En outre, les annotatrices ont rencontré une difficulté liée à l'interprétation axiologique, qui peut varier selon le locuteur. Il s'agit de la distinction entre les catégories de description et d'opinion (positive/négative). À titre d'exemple, la phrase « 고기만두라고 하지만, 두부와 야채를 많이 넣은 만두에요 (Ce sont des raviolis au bœuf, mais ils sont remplis de beaucoup de tofu et de légumes) » a été considérée comme relevant de la description par la première annotatrice (A1), qui a estimé qu'il s'agissait simplement d'un inventaire des farces des raviolis. En revanche, la seconde annotatrice (A2) a relié cette phrase à l'opinion négative, considérant que les raviolis au bœuf devaient être principalement farcis de viande et non de tofu ou de légumes. Cette tendance a également été observée dans le corpus HotelFR, dont la proportion de description était élevée par rapport à RestoFR (figures 1a, 1b). La distinction de ces catégories peut ainsi s'avérer quelque peu délicate, car si certains lecteurs acceptent la description pour ce qu'elle est, d'autres vont plus loin et la perçoivent comme reflétant un état favorable ou non d'après eux (l'opinion). Cela montre que l'évaluation dépend d'un système de normes et de valeurs propre à chaque locuteur. Ainsi, bien que chaque langue emploie des indices linguistiques différents pour exprimer leur point de vue axiologique, la typologie d'évaluation que nous avons élaborée montre qu'elle représente assez fidèlement ce en quoi consiste l'évaluation.

4 Conclusion

Notre conclusion principale est que la typologie que nous avons élaborée en nous appuyant sur le corpus RestoFR peut être appliquée à la fois à un autre domaine proche concernant l'expérience dans un lieu (l'hôtel) et à une autre langue (coréen). Cependant, la distinction entre l'opinion et la description présente des difficultés, car l'interprétation de ces deux catégories peut varier selon le locuteur. En outre, la détection automatique de HotelFR montre que des données annotées pour un domaine particulier peuvent s'appliquer à d'autres domaines comparables disposant de peu de données, ce qui permet de réduire les coûts d'annotation nécessaires pour couvrir le manque de données. En perspective, il serait intéressant de réaliser une annotation (et des modèles de classification) multilabels afin de résoudre les cas où les deux catégories d'évaluation se superposent. Par ailleurs, il serait avantageux d'augmenter la taille du corpus de référence de différents domaines et langues, et notamment de réaliser la détection automatique en langue coréenne en utilisant les modèles de langue pré-entraînés sur différentes langues comme XLM-RoBERTa (Conneau *et al.*, 2019).

Références

CARREIRAS M., GERNSBACHER M. A. & VILLA V. (1995). The advantage of first mention in spanish. *Psychonomic Bulletin & Review*, **2**(1), 124–129.

CHOLLET F. *et al.* (2015). Keras. https://github.com/fchollet/keras.

CLARK L. A. & WATSON D. (2016). Constructing validity : Basic issues in objective scale development. *Psychological Assessment*, **7**(3), 309–319.

COHEN J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, **20**(1), 37–46.

9. Nous reconnaissons néanmoins que ce résultat peut être dû à la petite taille du corpus.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv :1911.02116*.

ESHKOL-TARAVELLA I. & KANG H. J. (2019). Observation de l'expérience client dans les restaurants (Mapping Reviewers' Experience in Restaurants). In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN)-PFIA 2019-Volume II : Articles courts*, p. 361–370 : ATALA.

GERNSBACHER M. A. (1990). *Language Comprehension as Structure Building*. Psychology Press.

JURAFSKY D., CHAHUNEAU V., ROUTLEDGE B. & SMITH N. (2014). Narrative framing of consumer sentiment in online restaurant reviews. *First Monday*, **19**.

KANG H. J. & ESHKOL-TARAVELLA I. (2020). Les avis sur les restaurants à l'épreuve de l'apprentissage automatique (An Empirical Examination of Online Restaurant Reviews). In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, p. 249–257 : ATALA.

KERBRAT-ORECCHIONI C. (1980). *L'énonciation de la subjectivité dans le langage (4e édition) [version Kindle iOS]*. Armand Colin.

KIM S.-I., LEE J.-H. & GERNSBACHER M. A. (2004). The advantage of first mention in korean the temporal contributions of syntactic, semantic, and pragmatic factors. *Journal of psycholinguistic research*, **33**(6), 475–491.

LANDIS J. R. & KOCH G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, **33**(1), 159–174.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V. *et al.* (2011). Scikit-learn : Machine learning in python. *Journal of Machine Learning Research*, **12**, 2825–2830.

SZARVAS G., VINCZE V., FARKAS R., MÓRA G. & GUREVYCH I. (2012). Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, **38**(2), 335–367.

WACHSMUTH H., TRENKMANN M., STEIN B., ENGELS G. & PALAKARSKA T. (2014). A review corpus for argumentation analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, p. 115–127 : Springer.

Troisième partie

Résumés d'articles internationaux

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Extraction d'arguments basée sur les transformateurs pour des applications dans le domaine de la santé

Tobias Mayer[1] Elena Cabrio[1] Serena Villata[1]
(1) Université Côte d'Azur, CNRS, Inria, I3S, France
tmayer@i3s.unice.fr, {elena.cabrio,serena.villata}@univ.cotedazur.fr

RÉSUMÉ

Nous présentons des résumés en français et en anglais de l'article (Mayer *et al.*, 2020) présenté à la conférence 24th European Conference on Artificial Intelligence (ECAI-2020) en 2020.

ABSTRACT

Transformer-based Argument Mining for Healthcare Applications

We present French and English abstracts of the article (Mayer *et al.*, 2020) that was presented at the 24th European Conference on Artificial Intelligence (ECAI-2020).

MOTS-CLÉS : extraction d'arguments, essais contrôlés randomisés, PICO, prise de décision fondée sur des données probantes.

KEYWORDS: argument mining, randomized controlled trials, PICO, evidence-based decision making.

1 Résumé en français

L'extraction d'arguments (AM) vise généralement à identifier les composants argumentatifs dans le texte et à prédire les relations entre eux. La prise de décision basée sur les preuves (Hunter & Williams, 2012; Craven *et al.*, 2012; Longo & Hederman, 2013; Qassas *et al.*, 2015) en santé numerique vise à soutenir les cliniciens dans leur processus de délibération pour établir le meilleur plan d'action pour le cas en cours d'évaluation. Cependant, malgré son utilisation naturelle dans les applications de santé, seules quelques approches d'AM ont ete éppliqués à ce type de texte (Green, 2014; Mayer *et al.*, 2018, 2019), et leur contribution se limite à la détection des composants argumentatifs, sans prendre en compte la prédiction des relations entre eux. De plus, aucun grand ensemble de données annotées pour l'AM n'est disponible pour le domaine de la santé. Dans cet article (Mayer *et al.*, 2020), nous avons répondu à la question de recherche suivante : *comment définir un pipeline complet d'AM pour les essais cliniques ?* Pour répondre à cette question, nous proposons une approche basee sur des *transformers* bidirectionnels combinée à différents réseaux neuronaux pour la détection de composants argumentatifs et la prédiction de relations dans les essais cliniques, et nous évaluons cette approche sur un nouveau corpus de 659 résumés de la base de données MEDLINE. En particulier, nous avons étendu un jeu de données existant en annotant 500 résumés d'essais de MEDLINE, conduisant à 4198 composants argumentatifs et 2601 relations sur différentes maladies. En suivant les lignes directrices pour l'annotation des composants d'arguments dans les essais cliniques de (Trenta *et al.*, 2015), deux annotateurs ayant une formation en linguistique ont effectué l'annotation des 500 résumés sur le néoplasme. L'Inter Annotator Agreement (IAA) a été calculé sur 30 résumés (Fleiss

kappa : 0,72 pour les composants et de 0,68 pour la distinction preuves/conclusions) resultant dans un accord substantiel pour les deux tâches. Nous avons effectué l'annotation des relations argumentatives sur l'ensemble du corpus. L'IAA a été calculé sur 30 résumés annotés en parallèle par trois annotateurs (les deux mêmes annotateurs qui ont effectué l'annotation des composants argumentatifs, plus un annotateur supplémentaire), ce qui a donné un Fleiss kappa de 0,62. L'annotation des autres résumés a été effectuée par l'un des annotateurs susmentionnés. Nous avons proposé un pipeline complet d'extraction d'arguments pour les essais cliniques (les *preuves* et les *conclusions*), et prédisant les relations entre eux (*attaque* ou *support*). Plus précisément, notre pipeline complet d'AM pour les essais cliniques repose sur des transformateurs bidirectionnels profonds combinés à différents réseaux neuronaux, c'est-à-dire des Long Short-Term Memory (LSTM) networks, Gated Recurrent Unit (GRU) networks et Conditional Random Fields (CRFs). Nous avons procédé à une évaluation exhaustive de diverses architectures d'AM et nous avons obtenu un score F1 macro de 0,87 pour la détection des composants argumentatifs et de 0,68 pour la prédiction des relations. Notre évaluation a également révélé que les approches actuelles ne sont pas en mesure de relever adéquatement les défis posés par les textes médicaux et nous montrons que les approches basées sur les tranformers surpassent ces pipelines d'AM ainsi que les baselines standard.

2 English Abstract

Argument(ation) Mining (AM) typically aims at identifying argumentative components in text and predicting the relations among them. Evidence-based decision making (Hunter & Williams, 2012; Craven *et al.*, 2012; Longo & Hederman, 2013; Qassas *et al.*, 2015) in the healthcare domain targets at supporting clinicians in their deliberation process to establish the best course of action for the case under evaluation. However, despite its natural employment in healthcare applications, only few approaches have applied AM methods to this kind of text (Green, 2014; Mayer *et al.*, 2018, 2019), and their contribution is limited to the detection of argument components, disregarding the more complex phase of predicting the relations among them. In addition, no huge annotated dataset for AM is available for the healthcare domain. In this paper (Mayer *et al.*, 2020), we covered this gap, and we answered the following research question : *how to define a complete AM pipeline for clinical trials ?* To answer this question, we propose a deep bidirectional transformer approach combined with different neural networks to address the AM tasks of component detection and relation prediction in Randomized Controlled Trials, and we evaluate this approach on a new huge corpus of 659 abstracts from the MEDLINE database. In particular, we extended an existing dataset by annotating 500 abstracts of Randomized Controlled Trials (RCT) from the MEDLINE database, leading to a dataset of 4198 argument components and 2601 argument relations on different diseases. Following the guidelines for the annotation of argument components in RCT abstracts provided in (Trenta *et al.*, 2015), two annotators with background in computational linguistics carried out the annotation of the 500 abstracts on neoplasm. IAA among the annotators has been calculated on 30 abstracts, resulting in a Fleiss' kappa of 0.72 for argumentative components and 0.68 for the more fine-grained distinction between claims and evidence (meaning substantial agreement for both tasks). We carried out the annotation of argumentative relations over the whole dataset of RCT abstracts, including both the first version of the dataset (Trenta *et al.*, 2015) and the newly collected abstracts on neoplasm. IAA has been calculated on 30 abstracts annotated in parallel by three annotators (the same two annotators that carried out the argument component annotation, plus one additional annotator), resulting in a Fleiss' kappa of 0.62. The annotation of the remaining abstracts was carried

out by one of the above mentioned annotators. We proposed a complete argument mining pipeline for RCTs, classifying argument components as *evidence* and *claims*, and predicting the relation, i.e., *attack* or *support*, holding between those argument components. More precisely, we presented a complete AM pipeline for clinical trials relying on deep bidirectional transformers combined with different neural networks, i.e., Long Short-Term Memory (LSTM) networks, Gated Recurrent Unit (GRU) networks, and Conditional Random Fields (CRFs). We addressed an extensive evaluation of various AM architectures (e.g., for persuasive essays), and obtained a macro F1-score of .87 for component detection and .68 for relation prediction, outperforming current state-of-the-art end-to-end AM systems. Our evaluation also revealed that current approaches are unable to adequately address the challenges raised by medical text and we show that transformer-based approaches outperform these AM pipelines as well as standard baselines.

Références

CRAVEN R., TONI F., CADAR C., HADAD A. & WILLIAMS M. (2012). Efficient argumentation for medical decision-making. In *Proc. of KR 2012*, p. 598–602.

GREEN N. (2014). Argumentation for scientific claims in a biomedical research article. In *Proc. of ArgNLP 2014 workshop*.

HUNTER A. & WILLIAMS M. (2012). Aggregating evidence about the positive and negative effects of treatments. *Artificial Intelligence in Medicine*, **56**(3), 173–190. DOI : 10.1016/j.artmed.2012.09.004.

LONGO L. & HEDERMAN L. (2013). Argumentation theory for decision support in health-care : A comparison with machine learning. In *Proc. of BHI 2013*, p. 168–180.

MAYER T., CABRIO E., LIPPI M., TORRONI P. & VILLATA S. (2018). Argument mining on clinical trials. In *Proc. of COMMA 2018*, p. 137–148. DOI : 10.3233/978-1-61499-906-5-137.

MAYER T., CABRIO E. & VILLATA S. (2019). ACTA a tool for argumentative clinical trial analysis. In *Proc. of IJCAI 2019*, p. 6551–6553. DOI : 10.24963/ijcai.2019/953.

MAYER T., CABRIO E. & VILLATA S. (2020). Transformer-based argument mining for healthcare applications. In G. D. GIACOMO, A. CATALÁ, B. DILKINA, M. MILANO, S. BARRO, A. BUGARÍN & J. LANG, Éds., *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 de *Frontiers in Artificial Intelligence and Applications*, p. 2108–2115 : IOS Press. DOI : 10.3233/FAIA200334.

QASSAS M. A., FOGLI D., GIACOMIN M. & GUIDA G. (2015). Analysis of clinical discussions based on argumentation schemes. *Procedia Computer Science*, **64**, 282–289.

TRENTA A., HUNTER A. & RIEDEL S. (2015). Extraction of evidence tables from abstracts of randomized clinical trials using a maximum entropy classifier and global constraints. *CoRR*, **abs/1509.05209**.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Intégration de tâches: étiquetage morpho-syntaxique, analyse syntaxique et analyse sémantique traités comme une tâche unique

Timothée Bernard
Laboratoire de linguistique formelle, Université de Paris, France
timothee.bernard@u-paris.fr

RÉSUMÉ

Nous présentons des résumés en français et en anglais de l'article (Bernard, 2021), présenté lors de la conférence *16th Conference of the European Chapter of the Association for Computational Linguistics* (EACL 2021). L'article décrit l'intégration de tâches, un ensemble de principes orthogonaux au partage de paramètres dont le but est de maximiser l'interaction entre différentes tâches. L'intégration de tâches est illustrée avec un système analysant de manière jointe les niveaux morpho-syntaxiques, syntaxiques et sémantiques. La stratégie adoptée par ce système, entraîné par renforcement, est aussi analysée.

ABSTRACT

Multiple Tasks Integration: Tagging, Syntactic and Semantic Parsing as a Single Task

We present abstracts in English and in French for (Bernard, 2021), a paper was presented at the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021). The paper proposes Multiple Tasks Integration (MTI), a set of principles orthogonal to weight sharing the aim of which is to maximise the interaction between different tasks. MTI is illustrated with a system that performs part-of-speech tagging, syntactic dependency parsing and semantic dependency parsing. The strategy inferred by this system, trained by reinforcement learning, is also analysed.

MOTS-CLÉS : analyse sémantique, analyse syntaxique, étiquetage morpho-syntaxique, apprentissage multi-tâche, apprentissage par renforcement.

KEYWORDS: semantic parsing, syntactic parsing, POS tagging, joint processing, multitask learning, reinforcement learning.

1 Résumé en français

À la recherche d'alternatives autant aux chaînes de traitement séquentielles qu'aux systèmes mono-tâches, nous proposons l'intégration de tâches (MTI, pour *Multiple Tasks Integration*), un paradigme multi-tâche orthogonal au partage de paramètres. L'idée centrale de l'intégration de tâches est de traiter l'entrée simultanément sur différents niveaux d'analyse (i) de manière à ce que chaque décision repose sur l'ensemble des structures alors prédites et (ii) sans imposer aucune des contraintes d'ordre habituelles. De cette manière, les différentes tâches peuvent interagir pleinement les unes avec les autres. En particulier, nous ne forçons pas certains niveaux d'analyse à être analysés avant d'autres, et ni le début de la phrase avant la fin.

Volume 1 : conférence principale, pages 268–269.

Nous illustrons l'intégration de tâches avec un système analysant de manière jointe les niveaux morpho-syntaxiques, syntaxiques et sémantiques. L'intégration de tâches repose sur le calcul, à chaque étape, pour chaque token, d'un score pour chacune des actions possibles (sélections d'une étiquette morpho-syntaxique, sélection d'une tête syntaxique et d'une étiquette de dépendance, etc.) à partir d'un encodage de la phrase et des annotations déjà prédites. C'est à partir de ces scores que sont choisies les actions à appliquer, sans distinctions de type. Nous observons que la mise en place d'un apprentissage par renforcement ainsi que l'abandon des contraintes d'ordre mènent tous deux à un gain de performance sur les tâches d'analyse syntaxique et sémantique. Nous observons aussi que notre modèle adopte une stratégie de type *easy-first*, consistant — en moyenne — à prédire les dépendances en commençant par les plus courtes, mais que le niveau syntaxique n'est pas toujours analysé avant le niveau sémantique.

2 Abstract in English

Departing from both sequential pipelines and monotask systems, we propose Multiple Tasks Integration (MTI), a multitask paradigm orthogonal to weight sharing. The essence of MTI is to process the input iteratively but concurrently at multiple levels of analysis, where each decision is (i) based on all of the structures that are already inferred and (ii) free from usual ordering constraints. This way, the different tasks can fully interact with each other. In particular, we do not constrain the system to perform any given task before any other one, nor to analyse the beginning of the sentence before its end.

We illustrate MTI with a system that performs part-of-speech tagging, syntactic dependency parsing and semantic dependency parsing. The tasks integration is based on a scoring, at each step, for each token, of each possible actions (selection of a POS tag, selection of a syntactic head and a dependency label, etc.) using an encoding of both the sentence and the predictions from previous steps. The actions to perform are selected on the basis of theses scores, regardless of their type. We illustrate MTI with a system that performs part-of-speech tagging, syntactic dependency parsing and semantic dependency parsing. We observe that both the use of reinforcement learning and the release from sequential constraints are beneficial to the quality of the syntactic and semantic parses. We also observe that our model adopts an easy-first strategy that consists, on average, of predicting shorter dependencies before longer ones, but that syntax is not always tackled before semantics.

Références

BERNARD T. (2021). Multiple Tasks Integration : Tagging, Syntactic and Semantic Parsing as a Single Task. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 783–794, Online : Association for Computational Linguistics.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Modéliser la perception des genres musicaux à travers différentes cultures à partir de ressources linguistiques

Elena V. Epure[1] Guillaume Salha-Galvan[1, 2]
Manuel Moussallam[1] Romain Hennequin[1]
(1) Deezer Research, Paris, France
(2) LIX, École Polytechnique, Palaiseau, France
research@deezer.com

RÉSUMÉ

Nous résumons nos travaux de recherche, présentés à la conférence EMNLP 2020 et portant sur la modélisation de la perception des genres musicaux à travers différentes cultures, à partir de représentations sémantiques spécifiques à différentes langues.

ABSTRACT

Modeling the Music Genre Perception across Language-Bound Cultures

We summarize our research work, presented at the EMNLP 2020 conference, on modeling the music genre perception across cultures using language-specific semantic representations.

MOTS-CLÉS : annotation multilingue, multilinguisme, genres musicaux, représentations vectorielles multi-mots, ontologies, différences de perception interculturelles.

KEYWORDS: cross-lingual annotation, multilingualism, music genres, multi-word embeddings, ontologies, differences in perception across cultures.

1 Résumé en français

Des individus de cultures différentes peuvent utiliser différents genres musicaux pour décrire les mêmes artistes, albums ou morceaux de musique (Sordo *et al.*, 2008). Pour les plateformes de *streaming* musical comme Deezer, il est important de modéliser cette perception subjective des genres musicaux, afin d'améliorer la recherche et l'analyse d'informations musicales localisées ainsi que la recommandation personnalisée de contenu musical (Epure *et al.*, 2020a).

Dans ce travail de recherche, nous supposons que le terme de culture fait référence à des communautés partageant une langue commune et, partant de cette hypothèse, nous étudions la manière dont les entités musicales sont annotées en termes de genres musicaux selon différentes langues. Plus précisément, à partir des annotations d'un artiste, d'un album ou d'un morceau dans une langue source, nous cherchons à prédire les genres correspondants dans une langue cible. La perception des genres musicaux étant spécifique à chaque culture (Vela *et al.*, 2014; Blažytė & Liubinienė, 2016), modéliser ce problème d'annotation multilingue comme une tâche de traduction littérale est inenvisageable, car cette traduction ignorerait ces différences de perception. Par conséquent, nous proposons une approche différente reposant sur l'apprentissage de représentations sémantiques spécifiques à chaque langue, plus précisément des représentations vectorielles distribuées (de l'anglais « *distributed word*

embedding ») représentant des concepts multi-mots, ainsi que des ontologies musicales.

Nous évaluons notre approche pour six langues provenant de quatre familles différentes : germanique (anglais et néerlandais), romane (espagnol et français), japonique (japonais) et slave (tchèque). Nos expériences confirment qu'une traduction littérale des genres est inadaptée. Elles montrent qu'il est en revanche possible de réaliser avec précision une annotation multilingue non supervisée d'entités musicales en exploitant, comme proposé au sein de notre approche, les représentations vectorielles distribuées et les ontologies musicales dans les langues source et cible. Au sein de notre travail, nous discutons également les limites de l'hypothèse associant la notion de culture à celle de langue commune, qui semble forte pour certaines langues comme l'espagnol ou le français lorsque l'origine des annotateurs n'est pas précisée.

Enfin, nous rendons public le corpus multilingue utilisé au cours de nos expériences. Ce corpus peut être utile pour comparer différents modèles d'apprentissage de représentations vectorielles multilingues sur des données spécifiques au domaine musical. En outre, le cadre méthodologique proposé au sein de ce travail pourrait également être mis à profit pour étudier les différents comportements d'annotation à travers les cultures dans d'autres domaines tels que l'art.

Ce travail de recherche (Epure *et al.*, 2020b) a été publié dans les actes de la *2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2020).

2 English abstract

People from different cultures can refer to different music genres to describe the same music artists, albums or tracks (Sordo *et al.*, 2008). For music streaming services such as Deezer, it is crucial to model this subjective perception of music genres in order to improve localized music information retrieval and personalized music recommendation (Epure *et al.*, 2020a).

In this work, we assume that culture is related to a community speaking the same language and we study the cross-lingual music genre annotation of music. Namely, starting from music genre annotations of an artist, album or music track in a source language, we aim at predicting the corresponding music genres in a target language. Since music genres are Culture-Specific Items (Vela *et al.*, 2014; Blažytė & Liubinienė, 2016), modeling cross-lingual annotation as a literary translation task is unsuitable, as it could fail to account for cultural differences in music genre perception. Therefore, we propose a different approach relying on language-specific semantic representations : distributed embeddings for multi-word concepts and ontologies.

We evaluate our approach for six languages from four different language families : Germanic (English and Dutch), Romance (Spanish and French), Japonic (Japanese), and Slavic (Czech). Our experiments confirm that translation falls short for this task. On the other hand, we show that unsupervised cross-lingual music genre annotation is feasible with high accuracy when leveraging off-the-shelf distributed embeddings and ontologies in source and target languages. In our work, we also discuss the limits of relating a culture to a common language, which seems to be a strong assumption for some languages like Spanish and French when the origin of genre annotators is not specified.

We also release the cross-lingual corpus from our experiments, which could be used to benchmark multilingual pre-trained embedding models on domain-specific (namely music-specific) data. Furthermore, the proposed methodological framework could be leveraged to study the annotation behavior

across language-bound cultures in other domains too such as art.

This work (Epure *et al.*, 2020b) has been published in the proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020).

Références

BLAŽYTĖ D. & LIUBINIENĖ V. (2016). Culture-Specific Items (CSI) and their Translation Strategies in Martin Lindstrom's" Brand Sense". *Kalbų studijos*, (29), 42–57. DOI : 10.5755/j01.sal.0.29.16661.

EPURE E. V., SALHA G. & HENNEQUIN R. (2020a). Multilingual Music Genre Embeddings for Effective Cross-Lingual Music Item Annotation. In *Proceedings of the Twenty-First Conference of the International Society of Music Information Retrieval (ISMIR)*. https://archives.ismir.net/ismir2020/paper/000118.pdf.

EPURE E. V., SALHA G., MOUSSALLAM M. & HENNEQUIN R. (2020b). Modeling the Music Genre Perception across Language-Bound Cultures. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 4765–4779, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.386.

SORDO M., CELMA O., BLECH M. & GUAUS E. (2008). The Quest for Musical Genres : Do the Experts and the Wisdom of Crowds Agree ? In *Proceedings of the Ninth Conference of the International Society of Music Information Retrieval (ISMIR)*, p. 255–260. http://archives.ismir.net/ismir2008/paper/000267.pdf.

VELA M., SCHUMANN A.-K. & WURM A. (2014). Beyond Linguistic Equivalence. An Empirical Study of Translation Evaluation in a Translation Learner Corpus. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, p. 47–56, Gothenburg, Sweden : Association for Computational Linguistics. DOI : 10.3115/v1/W14-0308.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Revitalisation et Préservation des Langues Autochtones via le Traitement Automatique de la Langue Naturelle

Ngoc Tan Le Fatiha Sadat
Université du Québec à Montréal / Montréal, Québec, Canada
le.ngoc_tan@uqam.ca, sadat.fatiha@uqam.ca

RÉSUMÉ

Nous présentons des résumés en français et en anglais de l'article (Tan Le & Sadat, 2020) présenté à la 28ème conférence internationale sur les linguistiques computationnelles (*the 28th International Conference on Computational Linguistics*) en 2020.

ABSTRACT

Revitalization and Preservation of Indigenous Languages through Natural Language Processing

We present French and English abstracts of the article (Tan Le & Sadat, 2020) that was presented at the 28th International Conference on Computational Linguistics en 2020.

MOTS-CLÉS : Langues autochtones, inuktitut, langues peu dotées, prétraitement, segmentation morphologique, traduction automatique neuronale.

KEYWORDS: Indigenous languages, Inuktitut, low-resource languages, Preprocessing, Morphological Segmentation, Neural Machine Translation.

1 Résumé en français

Le traitement des langues autochtones a été très difficile au sein des tâches et des applications du TALN pour multiples raisons. En général, ces langues, dans la dimension linguistique, sont polysynthétiques et fortement infléchies avec une morphophonémie riche et des orthographes dialectales variables. De plus, les langues autochtones ont été considérées comme peu dotées et/ou en danger ; ce qui pose un grand défi pour la recherche liée à l'intelligence artificielle et à ses axes de recherche, dont le TALN.

Dans cet article [1], nous proposons une étude sur l'inuktitut afin de revitaliser et préserver la langue qui appartient à la famille inuite, parlée dans le nord du Canada. Nous nous concentrons sur : (1) le prétraitement, et (2) les applications sur des tâches du TALN spécifiques telles que l'analyse morphologique et la traduction automatique neuronale. Nos évaluations dans le contexte de la traduction automatique neuronale inuktitut-anglais peu dotée ont montré des améliorations significatives par rapport à l'état de l'art.

1. Présenté à Coling 2020 : https://www.aclweb.org/anthology/2020.coling-main.410/

2 English Abstract

Indigenous languages have been very challenging when dealing with NLP tasks and applications because of multiple reasons. These languages, in linguistic typology, are polysynthetic and highly inflected with rich morphophonemics and variable dialectal-dependent spellings ; which affected studies on any NLP task in the recent years. Moreover, Indigenous languages have been considered as low-resource and/or endangered ; which poses a great challenge for research related to Artificial Intelligence and its fields, such as NLP.

In this paper [2], we propose a study on the Inuktitut through pre-processing and neural machine translation, in order to revitalize the language which belongs to the Inuit family, spoken in Northern Canada. Our focus is concentrated on : (1) the preprocessing, and (2) applications on specific NLP tasks such as morphological analysis and neural machine translation. Our evaluations in the context of low-resource Inuktitut-English Neural Machine Translation, showed significant improvements of the proposed approach compared to the state-of-the-art.

Remerciements

The authors wish to express their thanks for professor Richard Compton, Canada Research Chair in Knowledge and Transmission of the Inuit Language, and the reviewers, for their constructive feedbacks.

Références

TAN LE N. & SADAT F. (2020). Revitalization of indigenous languages through pre-processing and neural machine translation : The case of Inuktitut. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 4661–4666, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.410.

2. Presented at Coling 2020 : https://www.aclweb.org/anthology/2020.coling-main.410/

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Simplification automatique de textes biomédicaux en français : les données précises de petite taille aident

Rémi Cardon[1] Natalia Grabar[1]
(1) CNRS, Univ. Lille, UMR 8163 - STL - Savoirs Textes Langage, F-59000 Lille, France
{remi.cardon, natalia.grabar}@univ-lille.fr

RÉSUMÉ

Nous présentons un résumé en français et un résumé en anglais de l'article (Cardon & Grabar, 2020), publié dans les actes de la conférence *28th International Conference on Computational Linguistics (COLING 2020)*.

ABSTRACT

French Biomedical Text Simplification : When Small and Precise Helps

We present a French abstract and an English abstract of the article (Cardon & Grabar, 2020), published in the proceedings of the 28th International Conference on Computational Linguistics (COLING 2020).

MOTS-CLÉS : simplification automatique de textes, domaine biomédical.

KEYWORDS: automatic text simplification, biomedical domain.

1 Résumé en français

Nous présentons des expériences de simplification automatique de textes biomédicaux en français. Nous travaillons au niveau de la phrase. Dans ce travail, nous utilisons deux corpus :

1. 4 596 couples de phrases parallèles extraites automatiquement à partir de corpus comparables du domaine de la santé en français (Cardon & Grabar, 2019),
2. 297 494 couples de phrases parallèles issues du corpus de langue générale WikiLarge (Zhang & Lapata, 2017), dédié à la simplification, que nous avons traduit automatiquement de l'anglais vers le français.

Pour effectuer la simplification automatiquement, nous utilisons l'outil OpenNMT-py (Klein *et al.*, 2017), créé à l'origine pour la traduction bilingue. Le fonctionnement de cet outil est basé sur une architecture encodeur-décoder avec mécanisme d'attention. Nous exploitons OpenNMT-py pour transformer un texte technique en un texte simplifié. Nous entraînons des modèles neuronaux sur les corpus parallèles constitués, en utilisant différents ratios de phrases de langue générale et spécialisée. En effet, nous avons un volume de phrases assez élevé pour décrire la simplification de la langue générale. Cependant, ces phrases ne décrivent pas bien les transformations requises pour simplifier la langue médicale. Les phrases parallèles provenant du domaine biomédical permettent donc de combler cette limite. Nous utilisons aussi un lexique qui apparie des termes médicaux complexes avec des paraphrases accessibles au grand public (7 580 paraphrases pour 4 516 termes médicaux). Nous pouvons ainsi mener trois séries d'expériences :

1. le lexique de paraphrases n'est pas utilisé et la simplification est uniquement basée sur les exemples provenant des corpus d'entraînement ;
2. le lexique est exploité lors de la phase de simplification, où il sert à indiquer au modèle comment remplacer les termes inconnus, qui se trouvent dans le lexique de paraphrases ;
3. le lexique est exploité lors de la phase d'entraînement, où il est ajouté à l'ensemble d'entraînement, ce qui permet de compléter les données des corpus.

Nous évaluons les résultats avec les métriques BLEU (Papineni *et al.*, 2002), SARI (Xu *et al.*, 2016) et Kandel (Kandel & Moles, 1958). Globalement, les résultats indiquent que des données spécialisées, même en petite quantité, aident significativement la simplification.

2 English Abstract

We present experiments on automatic biomedical text simplification in French. We work at the sentence level. In this work, we use two corpora :

1. 4 596 parallel sentence pairs automatically extracted from a French biomedical corpus (Cardon & Grabar, 2019),
2. 297 494 parallel sentence pairs obtained from general language corpus WikiLarge (Zhang & Lapata, 2017), which we have automatically translated from English to French.

In order to perform automatic simplification, we use the OpenNMT-py tool (Klein *et al.*, 2017). It was created for machine translation. This tool operates on an encoder-decoder architecture with an attention mechanism. We exploit OpenNMT-py to transform technical sentences into simpler sentences. We train neural models on the parallel corpora, using different ratios of general language and specialized language. Indeed, the volume of data is sufficient for describing general language simplification. Though, the sentences do not describe transformations that are specific to the medical domain. The parallel sentences from the medical domain allow us to fill this gap. We also use a lexicon that maps complex medical terms with laymen paraphrases (7 580 paraphrases for 4 516 medical terms). Thus we can perform three series of experiments

1. the lexicon is not used and the simplification is only based on the examples from the training corpora ;
2. the lexicon is exploited during the simplification phase, where it is used to indicate to the model how to substitute unknown terms that are present in the lexicon ;
3. the lexicon is exploited during the training phase, where it is added to the training set, where it complements the parallel corpora.

We evaluate the results with three metrics : BLEU (Papineni *et al.*, 2002), SARI (Xu *et al.*, 2016) and Kandel (Kandel & Moles, 1958). The results point out that little specialized data helps significantly the simplification.

Références

CARDON R. & GRABAR N. (2019). Parallel sentence retrieval from comparable corpora for biomedical text simplification. In *Proceedings of the International Conference on Recent Advances*

in Natural Language Processing (RANLP 2019), p. 168–177, Varna, Bulgaria : INCOMA Ltd. DOI : 10.26615/978-954-452-056-4_020.

CARDON R. & GRABAR N. (2020). French biomedical text simplification : When small and precise helps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, Barcelona, Spain (online) : Association for Computational Linguistics.

KANDEL L. & MOLES A. (1958). Application de l'indice de flesch à la langue française. *The Journal of Educational Research*, **21**, 283–287.

KLEIN G., KIM Y., DENG Y., SENELLART J. & RUSH A. M. (2017). OpenNMT : Open-source toolkit for neural machine translation. In *Proc. ACL*. DOI : 10.18653/v1/P17-4012.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318, Philadelphia, Pennsylvania, USA : Association for Computational Linguistics. DOI : 10.3115/1073083.1073135.

XU W., NAPOLES C., PAVLICK E., CHEN Q. & CALLISON-BURCH C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, **4**, 401–415. DOI : 10.1162/tacl_a_00107.

ZHANG X. & LAPATA M. (2017). Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 584–594, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/D17-1062.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Tabouid: un jeu de langage et de culture générale généré à partir de Wikipédia

Timothée Bernard
Laboratoire de linguistique formelle, Université de Paris, France
timothee.bernard@u-paris.fr

RÉSUMÉ

Nous présentons des résumés en français et en anglais de l'article (Bernard, 2020), présenté lors de la conférence *58th Annual Meeting of the Association for Computational Linguistics* (ACL 2020). L'article détaille comment un éventail de techniques relativement simples de TAL et d'apprentissage automatique peuvent être combinées pour générer à partir de Wikipédia le contenu d'un jeu de langage et de culture générale. L'article peut être vu comme définissant un projet stimulant pour des étudiant·e·s en TAL et le jeu lui-même a effectivement été implémenté sous la forme de Tabouid, une application Android et iOS.

ABSTRACT

Tabouid: a Wikipedia-based word guessing game

We present abstracts in English and in French for (Bernard, 2020), a paper that was presented at the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). The paper details how a range of relatively simple NLP and machine-learning techniques can be brought together to automically generate from Wikipedia the content of a word-guessing game. The paper can be seen as defining an engaging project for students enrolled in NLP programs and the game itself has been implemented as Tabouid, an Android and iOS application.

MOTS-CLÉS : jeu, Wikipédia, génération automatique de contenu, application, projet pédagogique.

KEYWORDS: game, Wikipedia, automatically generated content, application, pedagogical project.

1 Résumé en français

Nous présentons Tabouid, un jeu de langage et de culture générale généré automatiquement à partir de Wikipédia. Le but du jeu consiste à faire deviner aux autres joueurs et en un minimum de temps le titre d'une *carte* sans prononcer les mots interdits qu'elle spécifie. Tabouid contient 10 000 cartes (virtuelles) en anglais et autant en français. Ces cartes traitent non seulement de mots ou d'expressions courantes, mais aussi de concepts issus d'un large spectre de domaines tels que les arts, l'histoire ou les sciences. Chacune des cartes correspond à un article Wikipédia et, inversement, tout article peut servir à générer une carte. Un éventail de techniques relativement simples de TAL (racinisation, sélection par comptage, filtrage par règles, etc.) et d'apprentissage automatique (*scoring*, *active learning*, etc.) sont combinées efficacement pour former un algorithme en deux temps. Premièrement, une grande quantité d'articles Wikipédia sont évalués avec un score allant de 0 à 1 — ce score estime la difficulté, ou, vu autrement, la jouabilité de l'article. Ensuite, les meilleurs articles sont convertis en carte : il s'agit d'extraire, pour chacun d'entre eux, une liste de mots ou expressions interdites.

Volume 1 : conférence principale, pages 278–279.

Toutes les cartes de Tabouid étant associées à un score de jouabilité, la difficulté du jeu peut être réglée à l'aide d'un paramètre de seuil. Un tel réglage permet d'adapter le jeu à des publics divers, incluant par exemple enfants ou locuteurs non natifs, dont la maîtrise linguistique ou la connaissance de la culture associée à la langue sélectionnée peut varier. De plus, nous croyons que cet article à une certaine valeur pédagogique en définissant un projet d'implémentation pour des étudiant·e·s en TAL/linguistique computationnelle.

Le jeu est disponible sous la forme d'une application Android (https://play.google.com/store/apps/details?id=com.tot.tabouid) et iOS (https://apps.apple.com/us/app/tabouid/id1477994156) entièrement gratuite et ne contenant aucune publicité. Les cartes ont été pré-générées et incluses dans l'application ; aucune connexion internet n'est requise durant le jeu.

2 Abstract in English

We present Tabouid, a word-guessing game automatically generated from Wikipedia. During a turn of the game, one player tries to make the other players guess the title of *card* as quickly as they can and without using any of the banned words mentionned on the card. Tabouid contains 10,000 (virtual) cards in English, and as many in French, covering not only words and linguistic expressions but also a variety of topics including artists, historical events or scientific concepts. Each card corresponds to a Wikipedia article, and conversely, any article could be turned into a card. A range of relatively simple NLP (stemming, count-based selection, rule-based filtering, etc.) and machine-learning (scoring, active learning, etc.) techniques are effectively integrated into a two-stage process. First, a large subset of Wikipedia articles are scored — this score estimates the difficulty, or alternatively, the playability of the page. Then, the best articles are turned into cards by selecting, for each of them, a list of banned words based on its content.

All cards in Tabouid are associated with a difficulty score. This allows the difficulty level of the game to be set in a straightforward way. With such an adaptable difficulty, the game can accommodate various groups of players, which could include individuals such as children or foreigners, whose level of proficiency or knowledge of the culture associated with the target language may vary. In addition, we believe that this paper can have some pedagogical value by defining an implementation project for students enrolled in NLP or computational linguistic programs.

The game is available as an Android (https://play.google.com/store/apps/details?id=com.tot.tabouid) and iOS (https://apps.apple.com/us/app/tabouid/id1477994156) application that is entirely free and does not contain any advertisement. The cards have been pre-generated and packaged with the application; an internet connection is not required during the game.

Références

BERNARD T. (2020). Tabouid : a Wikipedia-based word guessing game. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, p. 24–29 : Association for Computational Linguistics.