Traitement Automatique des Langues Naturelles (TALN) [1]

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles.
Volume 2 : 23e REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL)

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

Communications orales

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

Améliorer un agent conversationnel : prendre en compte à la volée des retours utilisateurs

Maxime Arens[1,2]
(1) IRIT, Cours Rose Dieng-Kuntz, 31400 Toulouse, France
(2) Synapse Développement, 7 Boulevard de la Gare, 31500 Toulouse, France
maxime.arens@irit.fr

RÉSUMÉ

Nous présentons une approche améliorant la pertinence des réponses d'un système conversationnel de question-réponse en profitant de l'expérience passée du système. Un agent conversationnel déployé au contact d'utilisateurs peut en effet profiter de retours afin d'améliorer la validité de ces futures réponses. Les systèmes de question-réponse fonctionnent généralement autour d'un modèle rapprochant sémantiquement une question à une ou plusieurs réponses potentielles. Ici, nous prenons en compte le cas où le modèle de correspondance rapproche une question à une liste de réponses associées à un score de pertinence. Une approche classique pour prendre en compte les retours d'utilisateurs, est de les utiliser pour augmenter le nombre de données de réentrainement du modèle de rapprochement sémantique. Nous proposons une approche différente, impactant le score des réponses potentielles, où nous prenons en compte « à la volée » les retours utilisateurs : entre le moment où l'utilisateur pose une nouvelle question et celui où le système lui répond.

ABSTRACT

Improve a conversational agent : considering on the fly user feedback.

We present an approach to improve the relevance of a conversational question answering system by leveraging previous user feedback. A dialog system deployed in contact of users can take into accounts feedbacks to improve the relevance of its answers. Question answering systems usually work through models matching a question with one or multiple answers. Here we consider the case where the model matches a question to a list of answers scored by relevance. A classical approach of considering user feedback is to augment the training data used to retrain the matching model. Here we suggest a different approach, impacting answers scores, by considering "on the fly" the feedbacks : between when the user asks a new question and when the system responds.

MOTS-CLÉS : Question-réponse conversationnelle ; Retours utilisateurs ; Similarité entre questions ; Apprentissage actif.

KEYWORDS: Conversational question answering ; User feedback ; Question similarity ; Active Learning.

Volume 2 : 23e REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL), pages 2–14.

1 Introduction

Les systèmes conversationnels de question-réponse permettent, au fil d'une conversation, suite à une question formulée sous la forme d'une requête en langage naturel, de retourner une réponse issue d'une base de connaissances (Reddy *et al.*, 2019). De telles réponses peuvent être générées à partir d'informations ou bien, comme dans le cas nous concernant, extraites d'un corpus de document (Hoi *et al.*, 2018). Ces agents mêlent à la fois des techniques issues de la discipline du Traitement Automatique des Langues (TAL) et de celle de la Recherche d'Information (RI) (Belkin *et al.*, 1995). En étudiant le fonctionnement de ces systèmes d'un point de vue chronologique, la compréhension de la requête utilisateur (Qu *et al.*, 2019) est plus précisément une tâche de Compréhension du Langage Naturel (sous-branche du TAL), tandis que l'identification du document contenant la réponse et son extraction (Tellex *et al.*, 2003) appartiennent plus au domaine de la RI.

Certains agents évoluent sur des domaines très ouverts et grand public (Rajpurkar *et al.*, 2016; Qu *et al.*, 2020) tandis que d'autres se focalisent sur des domaines restreints et techniques (Campos *et al.*, 2020b). La démocratisation de ces systèmes au sein des entreprises, en tant qu'outil de support de la relation client ou bien à des fins internes de gestion de ressources informatives (Gao *et al.*, 2019), rend l'adaptation de l'agent au domaine de l'entreprise souvent nécessaire. Cette spécialisation, nécessaire pour élaborer des systèmes conversationnels (Aliannejadi *et al.*, 2019) portant sur des sujets précis et à haute technicité, nécessite des données d'entraînement (Campos *et al.*, 2020b). Ces données d'entraînement sont souvent des données annotées manuellement par des experts (par exemple par l'annotation d'une réponse correcte pour une certaine réponse). Or, le recours à des experts, particulièrement sur des domaines spécialisés et techniques, est onéreuse.

Un véritable enjeu existe donc pour limiter le recours à ces experts afin de réduire les ressources nécessaires à l'adaptation de ces systèmes. Une façon de répondre à cet enjeu est de déployer un système conversationnel partiellement spécialisé, de le faire interagir avec des utilisateurs puis de prendre en compte leurs retours pour améliorer la performance de l'agent (Hancock *et al.*, 2019). Les retours utilisateurs peuvent évaluer chaque tour de la conversation individuellement ou donner une appréciation globale de la conversation. Nous utilisons le cas où l'utilisateur évalue des tours de conversation, plus précisément, évalue binairement (positivement ou négativement) la réponse du système à sa question.

Puisque ici le système conversationnel repose sur une interaction avec des agents humains (Li *et al.*, 2016) pour son entraînement, il est important que les mécanismes d'apprentissage du système prennent en compte certaines particularités de ce cas d'utilisation réelle. Tout d'abord, les requêtes d'utilisateurs humains, bien que véhiculant parfois le même questionnement, sont souvent formulées de manières différentes et peuvent contenir des fautes (Christmann *et al.*, 2019). Ensuite, les retours des utilisateurs étant par définition subjectifs, des désaccords peuvent naître autour de la perception de la qualité d'une réponse apportée par le système. De plus, les utilisateurs peuvent volontairement ou involontairement donner des retours négatifs sur une réponse qu'un expert jugerait correcte. Enfin, le niveau d'amélioration du système est dépendant du nombre, de la qualité et de la portée des retours utilisateurs.

Une approche classique pour améliorer un système conversationnel à partir de retours utilisateurs, est de se servir de ces retours comme données supplémentaires lors d'un réentrainement du modèle rapprochant une question à des réponses (Campos *et al.*, 2020a). Cette approche nécessite tout d'abord que le module de rapprochement sémantique question-réponse soit réentrainable et qu'une boucle de réentrainement/redéploiement du module soit implémentée (Liu *et al.*, 2018). Nous présentons

ici une approche différente permettant de s'affranchir de ces deux conditions. On suppose que le module de rapprochement sémantique retourne une liste de réponses ayant chacune un score de pertinence. Tout d'abord, au moment où l'utilisateur pose sa question, nous récupérons grâce à un modèle d'équivalence entre questions le plus grand nombre de retours utilisateurs liés à des réponses apportées par le système pour des questions équivalentes (Prabowo & Budi Herwanto, 2019). Nous formons alors des quadrets comprenant chacun une question équivalente à la question source, une réponse apportée par le système et les retours utilisateurs binaires sur ce couple question-réponse. Grâce à une fonction prenant en entrée : le score originel des réponses envisagées pour la question et les retours d'utilisateurs évaluant ces réponses par rapport à des variantes passées de sa question, l'algorithme que nous présentons calcule un nouveau score pour chacune des réponses potentielles. Enfin, le système retourne à l'utilisateur la réponse ayant le meilleur score de pertinence, après ajustement du score de pertinence au moyen de la prise en compte de ses expériences passées.

2 Approche proposée

Dans cette section nous allons détailler l'approche que nous proposons dans cet article. Nous commencerons par expliciter l'ensemble du processus d'un point de vue général. Ensuite, nous aborderons en détail notre choix de modèle d'équivalence et son fonctionnement. Enfin, nous discuterons de la fonction modifiant le score des réponses en fonction des retours utilisateurs.

2.1 Architecture

Lorsqu'un utilisateur pose une question, le système obtient une liste de réponses potentielles et retourne à l'utilisateur, celle ayant le score de pertinence le plus élevé. Suite à cet échange, l'utilisateur peut ensuite évaluer la réponse du système en la jugeant satisfaisante ou non satisfaisante. Ce retour est matérialisé par une mise en mémoire en base de données du quadret (question utilisateur ; réponse du système ; nombre de retours utilisateurs positifs ; nombre de retours utilisateurs négatifs). La question utilisateur est donc la requête faite par l'utilisateur en langage naturel. La réponse du système est un extrait d'un document faisant partie de la base de connaissances de l'agent conversationnel. Finalement, le retour utilisateur est une valeur binaire : 0 pour une réponse n'ayant pas satisfait l'utilisateur, 1 pour une réponse l'ayant satisfait. On note que de par le fonctionnement des bases de données, deux questions utilisateurs non identiques (avec une faute d'orthographe par exemple) constitueront deux lignes différentes dans la table de données, leurs nombres de retours utilisateurs ne seront donc pas rassemblés dans la base.

L'approche que nous proposons consiste en l'ajout d'un processus entre le moment où l'utilisateur pose sa question et le système lui répond. Ce processus est constitué de plusieurs étapes. Pour commencer, nous récupérons la liste des réponses envisagées par le système suite à la question de l'utilisateur. Pour chacune de ces réponses, nous collectons en base de données la liste des quadrets contenant les questions utilisateurs ayant fait remonter cette réponse ainsi que les retours faits par les utilisateurs sur leur satisfaction liée à cette réponse pour leur question. À l'aide du modèle d'équivalence entre questions, le système mis en oeuvre compare ensuite la nouvelle question utilisateur avec chacune des questions utilisateurs contenues dans les listes de quadrets. Le système identifie donc quels sont les quadrets contenant des retours utilisateurs sur la pertinence des réponses à des variantes de la question posée par l'utilisateur. Pour chacune des réponses contenues dans la

liste des réponses envisagées nous avons donc potentiellement une liste de quadrets contenant de l'information pertinente. Le système itère alors sur la liste de réponses potentielles afin de modifier le score de pertinence de cette réponse grâce aux nouvelles informations pertinentes obtenues. Cette fonction de modification prend le score de pertinence initial, les sommes des retours utilisateurs positifs et négatifs associés à cette réponse, et retourne un nouveau score de pertinence. Nous trions de nouveau la liste de réponse pour prendre en compte les potentiels changements de classement entre les réponses. Enfin, nous renvoyons à l'utilisateur la réponse en haut du classement, celle qui a le plus haut score de pertinence.

2.2 Similarité entre questions

Afin d'expliquer le fonctionnement du module d'équivalence entre questions, nous commencerons par discuter de ces entrées et des ces sorties. En entrée, nous avons la question posée par l'utilisateur et une liste de questions utilisateurs (le même ou d'autres utilisateurs) posées par le passé au système. En sortie, nous avons les indices des questions utilisateurs passées, considérées comme étant des variantes de la question que vient de poser l'utilisateur. Nous entendons par variante, une question ayant le même sens, contenant des fautes d'orthographe ou étant une reformulation de la question posée par l'utilisateur.

Cette détection d'équivalence de questions est une étape critique du processus proposé. En effet, étant située entre le moment où l'utilisateur pose sa question et obtient sa réponse, cette détection ne peut pas durer plus d'un certain temps si on ne veut pas impacter le ressenti de l'utilisateur. De plus, la précision du modèle est un critère très important. Dans le cas où, le modèle se tromperait en classifiant une question comme équivalente, le système global prendrait alors en compte des retours d'utilisateurs qualifiant un couple question-réponse n'ayant rien à voir avec celui qu'il essaie d'évaluer.

Avec ces deux idées en tête, nous avons entraîné un classifieur permettant d'identifier une question comme variante d'une autre. Nous sommes partis sur la piste de prendre des modèles évaluant la similarité entre deux phrases (Agirre *et al.*, 2012), puis de spécialiser ces modèles sur la tâche nous intéressant. Premièrement nous avons constitué un corpus de données pour l'entraînement et l'évaluation de tels modèles. Afin d'être au plus proche de notre cas d'utilisation réelle, nous avons récupéré l'ensemble des requêtes utilisateurs faites à un agent conversationnel lors de son déploiement. Cet agent est destiné au grand public sur le domaine de l'entrepreneuriat en France. Nous avons combiné entre elles les différentes requêtes utilisateurs afin de former des couples questions utilisateurs. À l'aide d'un modèle de similarité entre phrases, nous avons ensuite formé un ensemble de données plus petit composé de couples de questions potentiellement similaires. Le corpus, ainsi obtenu, atteint un peu plus de 4 000 couples de questions. Deux experts ont alors annotés à la main chacun de ces couples comme étant ou non des variantes d'une même question. Grâce à une étape de réconciliation, les experts se sont accordés sur le label de chaque couple.

Une fois le corpus obtenu, l'étape suivante a été de réfléchir sur les caractéristiques des questions sur lesquelles nos modèles allaient s'appuyer pour faire leur classification. En effet, ces modèles fonctionnent en calculant la distance [1] entre les représentations distribuées (plongements lexicaux) des mots composants les questions (Kusner *et al.*, 2015). Le temps d'exécution du calcul des caractéristiques des questions pouvant être trop lent pour les applicatifs visés, nous avons essayé un

1. Le calcul des distances mentionnées dans la Table 1 est réalisé entre la moyenne des vecteurs représentatifs des mots de chacune des questions.

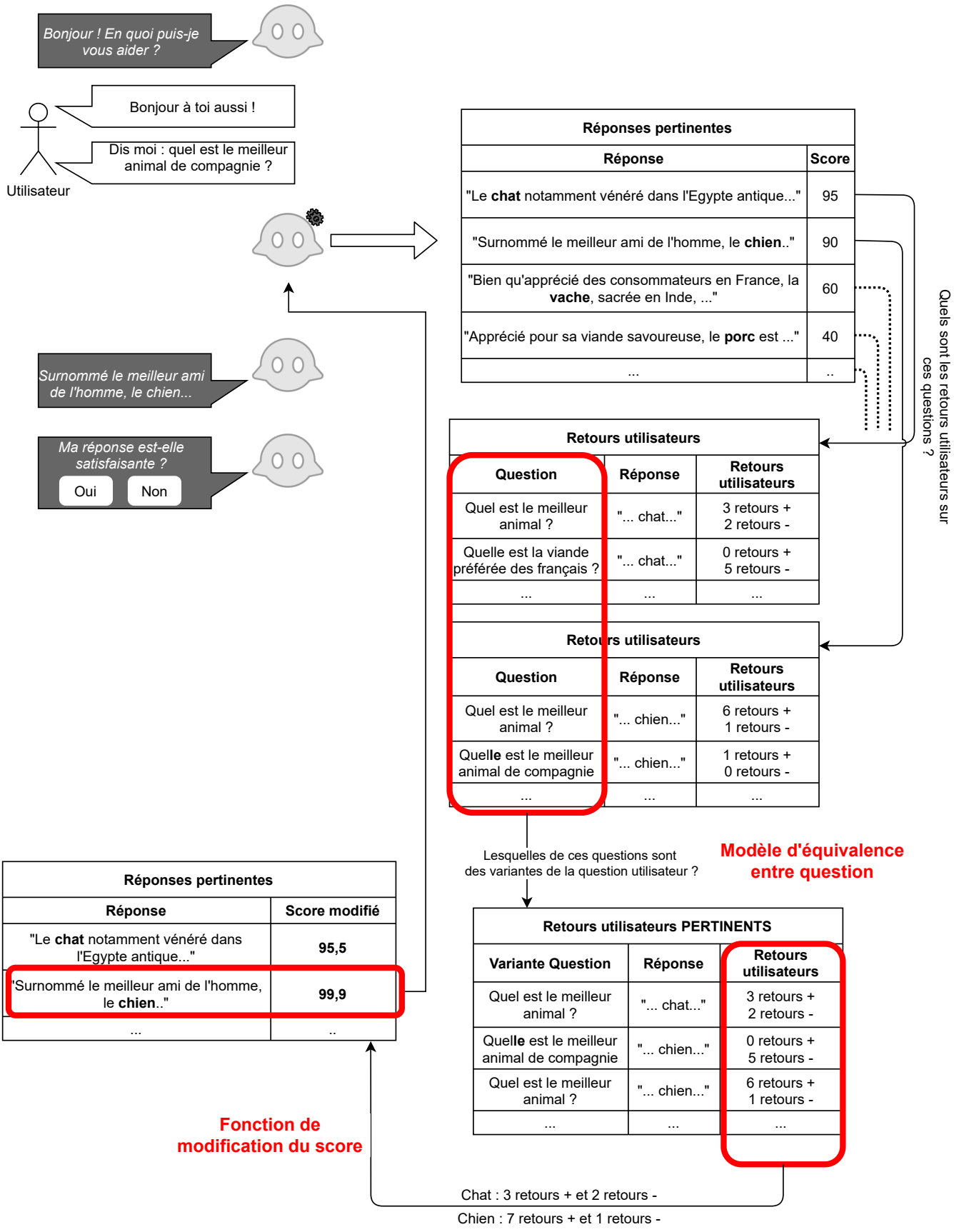


FIGURE 1 – Exemple d'exécution de l'algorithme proposé

important spectre de caractéristiques : certaines simples à calculer et d'autres un peu plus complexes à obtenir.

Caractéristiques simples	Caractéristiques intermédiaires	Caractéristiques complexes
Nombre de mots en commun	Taille de la plus grande sous-phrase en commun	Distance cosinus
Nombre de mots total	Pourcentage de tokens ayant du sens en commun	Distance de Manhattan
Pourcentage de mots en commun	Pourcentage de tokens vide de sens en commun	Distance de Canberra
Premier mot identique	Distance de Levenshtein	Distance euclidienne
Dernier mot identique		Distance de Minkowski

TABLE 1 – Tableau des caractéristiques essayées

Nous avons ensuite évalué différents classifieurs, avec certaines des caractéristiques de la Table 1 afin de trouver celui combinant les meilleures performances en matière de f1-score et de temps d'exécution. Les algorithmes derrière ces classifieurs reposent sur la minimisation de la différence entre les différentes valeurs caractéristiques des questions listées dans la Table 1.

Noms des méthodes
Recherche des plus proches voisins
Gradient stochastique (Bottou, 2010)
Forêt d'arbres décisionnels
Régression logistique
Machines à vecteurs de support
XGBoost (Chen & Guestrin, 2016)

TABLE 2 – Tableau des méthodes de classification évaluées

Suite aux évaluations présentées dans la prochaine section de cet article, nous avons donc choisi un classifieur reposant sur l'algorithme XGBoost répondant à nos critères de précision et de performance temporelle.

2.3 Modification du score

Une grande partie de l'efficacité de l'approche proposée dans cet article est liée à la fonction modifiant le score de pertinence des réponses potentielles en fonction des retours utilisateurs passés. Cette fonction doit répondre à certaines exigences propres à l'application industrielle à laquelle elle prend part et avoir certains comportements. Les scores doivent rester bornés entre 0 et 100 afin de s'accorder avec le fonctionnement de l'agent conversationnel que nous cherchons à améliorer. Dans le cas où il y a le même nombre de retours utilisateurs positifs et négatifs sur une réponse, le score de la réponse ne doit pas être modifié, car nous considérons alors qu'il n'y a pas d'accord entre les évaluateurs et que de ce fait leurs retours sont difficilement exploitables. Un vote utilisateur doit avoir un impact important sur le score de la réponse afin de pouvoir faire remonter le plus vite possible une bonne réponse en première position. Enfin, la fonction doit prendre en compte l'accord entre les utilisateurs sur la pertinence de cette réponse. L'impact sur le score ne doit pas être le même pour une réponse où les utilisateurs sont majoritairement d'accord, et une réponse provoquant un désaccord (un grand nombre de retours utilisateurs en opposition).

Le $NouveauScore$ est défini par :

si $DiffRetours \leq 0$, $NouveauScore = \frac{AncienScore}{e^{-Accord*DiffRetours}}$

sinon, $NouveauScore = 100 - 2 * (100 - AncienScore) + \frac{2*(100-AncienScore)}{1+e^{-Accord*DiffRetours}}$

avec r_+ = nombres de retours positifs, r_- = nombres de retours négatifs,

$$Accord = \frac{|r_+ - r_-|}{r_+ + r_-}$$

$$DiffRetours = r_+ - r_-$$

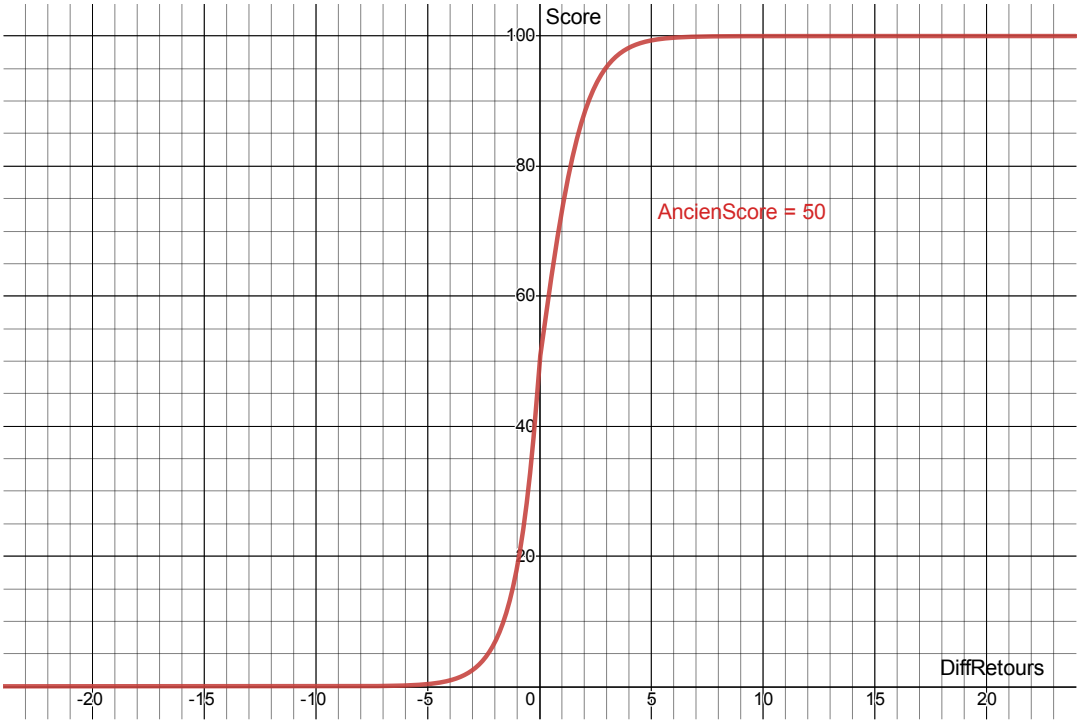


FIGURE 2 – Exemple de la fonction de modification du score

Comme l'illustre la Figure 2, les retours de cette fonction sont bien bornés entre 0 et 100. La nature exponentielle de la fonction induit bien que les premiers votes utilisateurs impactent fortement les scores. Une réponse jugée pertinente par les utilisateurs verra son score réévalué à la hausse et une réponse jugée non pertinente verra son score réévalué à la baisse. Si les réponses apportées par le système sont jugées peu pertinentes pour une question donnée, en ayant dégradé le score associé à ces réponses, le système présentera alors les réponses qui suivent dans la liste ordonnée qu'il produit. La liste des réponses potentielles sera ainsi progressivement parcourue. La fonction contient bien une identité, ne modifiant pas le score d'une réponse s'il n'y a pas de différence entre le nombre de retours positifs et négatifs qu'elle a reçue (ou si elle n'a reçue aucun retour utilisateur).

Le coefficient nommé « accord » impacte la pente de cette fonction : plus l'écart est faible entre un grand nombre de retours positifs et négatifs, moins la fonction d'activation est pentue et donc moins ces retours impactent le score de la réponse. Le parti pris étant de ne pas encourir le risque de modifier le score d'une réponse suscitant un important désaccord entre les utilisateurs.

Notons que la fonction a été divisée en deux parties pour des raisons de simplification de gestion au niveau de l'implémentation de l'agent conversationnel sur lequel s'est greffé l'approche proposée dans cet article.

3 Évaluation

Étant donné que l'approche proposée dans cet article s'inscrit dans une démarche scientifique et est aussi déployée dans des conditions d'utilisation réelles, nous avons dû évaluer certains aspects de la méthode. Dans un premier temps, nous avons dû vérifier le fait que l'expérience de l'utilisateur n'était pas dégradée. Ensuite, nous avons dû entraîner et sélectionner le classifieur de questions équivalentes le plus performant possible. Enfin, nous avons évalué les gains de cette approche.

3.1 Performance temporelle

L'approche proposée rajoutant un processus entre le moment où l'utilisateur pose sa question et celui où il obtient sa réponse, la durée d'exécution du processus est un paramètre critique. Nous avons tout d'abord identifié quelles étapes pouvaient être chronophages. La fonction changeant le score étant relativement simple d'un point de vue mathématique, ce n'est pas une étape prenant du temps. Le module retournant les questions similaires est déployé sous forme de service pouvant être appelé par l'agent conversationnel. Après évaluation, nous considérerons que le temps de réponse de ce service est égal à la somme du temps d'extraction des caractéristiques des questions et du temps d'exécution du classifieur.

Après avoir optimisé la durée d'exécution de cette étape nous avons réalisé une évaluation du temps de réponse de l'agent conversationnel. Pour cela nous avons posé au système les 10 questions les plus fréquemment utilisées par les utilisateurs et avons réalisé la moyenne des temps de réponses du système. Le temps de réponse moyen sur ces 10 réponses est d'environ 936 ms. Le temps de réponse maximal durant cette évaluation est de 1026ms. Le critère du temps de réponse maximal de l'agent conversationnel étant fixé à 2s, nous pouvons considérer que notre approche ne nuit pas à l'expérience de l'utilisateur en terme de temps d'attente de réponse.

3.2 Modèle d'équivalence entre questions

Dans le but d'évaluer le meilleur classifieur, nous avons évalué les performances des modèles suite à leurs entrainements. Nous avons pu évaluer la précision, le rappel et le f1-score de nos classifieurs sur le sous-ensemble de données annotées par les experts que nous avions gardé pour l'évaluation. Ces métriques sont évaluées pour chacune des classes prédites par nos modèles : questions équivalentes et non-équivalentes. Nous rappelons que le critère de sélection est le f1-score, rapport entre la précision et le rappel, sur la classe des questions équivalentes.

Nom du modèle	Classe	Précision	Rappel	F1-Score
Gradient stochastique	non-équivalentes	0.83	0.96	0.89
Gradient stochastique	équivalentes	0.89	0.60	0.72
Régression logistique	non-équivalentes	0.87	0.97	0.92
Régression logistique	équivalentes	**0.93**	0.7	0.80
Machines à vecteurs de support	non-équivalentes	0.93	0.94	0.93
Machines à vecteurs de support	équivalentes	0.87	0.85	0.86
Extreme Gradient Boosting	non-équivalentes	0.97	0.94	0.96
Extreme Gradient Boosting	équivalentes	0.88	**0.95**	**0.92**

TABLE 3 – Evaluation des classifieurs

Le classifieur proposant les meilleurs performances est un classifieur basé sur l'algorithme XGBoost (Chen & Guestrin, 2016) utilisant une grande majorité des caractéristiques de questions que nous avions explorées. Les performances de ce classifieur sont particulièrement élevées dans la détection des petites variations entre questions comme les fautes d'orthographe ou les fautes de frappe. Notons que le classifieur propose aussi le meilleur f1-score sur la classe des questions non-équivalentes même si cela n'impacte pas directement l'approche que nous proposons.

Remarquons que le classifieur proposant la meilleure précision sur la classe des questions équivalentes est celui basé sur une régression logistique. Nous ne l'avons pas retenu car son taux de rappel était trop faible.

Nous avons aussi exploré des modèles neuronaux qui retournaient de bons résultats, mais le gain en performance ne justifiait pas la perte au niveau du temps d'exécution.

3.3 Gain de l'approche

Le travail rapporté dans cet article est le fruit d'une réflexion ayant eu pour objectif d'améliorer les performances d'agents conversationnels déjà déployés auprès d'utilisateurs. Antérieurement à ce travail, les retours utilisateurs étaient pris en compte de façon rudimentaire, n'impactant qu'à la marge la qualité des réponses apportées par le système conversationnel. En effet, les retours utilisateurs n'étaient pris en compte que pour les questions posées précédemment de façon identique. Ainsi, une erreur de typographie empêche toute récupération des retours utilisateurs précédents. De plus, un retour utilisateur ne faisait que incrémenter (retour positif) ou décrémenter (retour négatif) de 1 le score de la réponse.

Afin d'évaluer le gain de notre approche nous avons récupéré un corpus créé par des experts du domaine de connaissance sur lequel évolue le système. Ce corpus comprend plus de 250 questions techniques propres au domaine de l'entreprenariat et la liste des réponses que retourne le système. Pour chacune des questions la position de la réponse attendue dans cette liste est de plus renseignée quand c'est possible par les experts.

En partant de l'hypothèse que les retours d'utilisateurs sont toujours corrects, nous avons simulé l'évolution de la précision moyenne (ici le pourcentage de questions issues du corpus où l'agent propose la bonne réponse directement à l'utilisateur) de l'agent conversationnel au fil des retours utilisateurs.

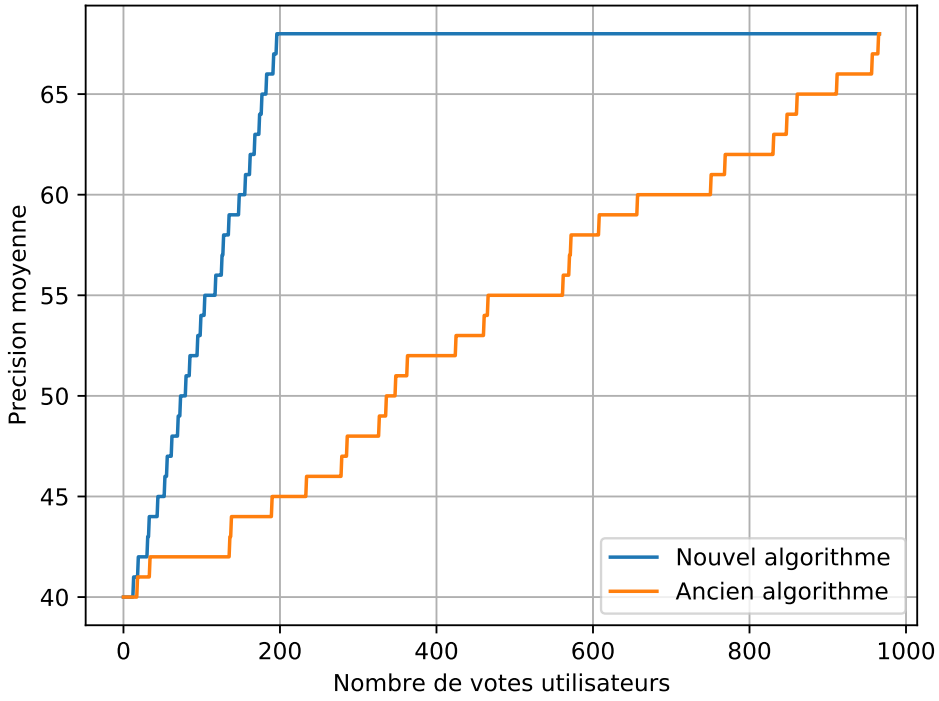


FIGURE 3 – Comparaison de la précision moyenne de l'agent conversationnel avec ou sans l'approche proposée

Comme nous pouvons le voir sur les résultats ci-dessous, notre approche permet de faire remonter en première position les bonnes réponses, améliorant la précision de l'agent, et cela beaucoup plus rapidement que dans l'approche précédemment utilisée.

4 Conclusion

Dans cet article, nous avons proposé une approche permettant d'améliorer les performances d'un agent conversationnel après son déploiement. En effet, lorsqu'un utilisateur pose une question au système, nous prenons en compte les retours que les utilisateurs ont fait sur des réponses à des questions équivalentes. Nous proposons donc la réponse la plus pertinente en utilisant l'expérience accumulée par l'agent.

Cette prise en compte des retours utilisateurs se fait « à la volée » entre le moment où l'utilisateur pose sa question et le moment où le système lui répond. L'approche peut se greffer par dessus tout type d'algorithme de rapprochement sémantique et fonctionne donc même quand cet algorithme ne prévoit pas de ré-entraînement. De plus, dans le système proposé les retours utilisateurs l'impactent immédiatement. Tandis que dans les approches où les systèmes se ré-entraînent, la prise en compte d'un retour utilisateur ne se fait qu'au moment du prochain ré-entraînement.

L'approche proposée utilise un modèle de détection d'équivalence entre questions. Quand un uti-

lisateur pose une question, l'agent peut récupérer des retours sur des réponses proposées pour des variantes de cette question. Cela permet au système de profiter encore plus de son expérience.

L'amélioration du système est liée à la qualité des retours des utilisateurs. En cas de désaccord entre les utilisateurs, nous avons choisi d'atténuer leur impact sur la réponse du système. Nous pouvons imaginer une approche complémentaire, dans laquelle un superviseur expert viendrait départager le désaccord. Dans d'autres approches, un désaccord entre utilisateurs pourrait brouiller l'apprentissage en accumulant des exemples contradictoires.

Finalement, nous pensons qu'il serait intéressant d'identifier au moment de la récupération des questions équivalentes, les couples questions-réponses ayant un nombre conséquent de retour, mais dans des directions opposées. Par exemple, un couple ayant 5 retours positifs et un autre ayant 7 retours négatifs. Nous pouvons considérer que l'accord fort entre les utilisateurs sur ces couples signifie que la réponse évaluée est la bonne pour l'une des questions mais une mauvaise pour l'autre. Or ces deux questions ont été classifiées comme étant équivalentes, une variante de la même question. Nous pourrions utiliser ces contre-exemples afin d'améliorer notre classifieur.

Références

AGIRRE E., CER D., DIAB M. & GONZALEZ-AGIRRE A. (2012). SemEval-2012 task 6 : A pilot on semantic textual similarity. In **SEM 2012 : The First Joint Conference on Lexical and Computational Semantics – Volume 1 : Proceedings of the main conference and the shared task, and Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, p. 385–393, Montréal, Canada : Association for Computational Linguistics.

ALIANNEJADI M., ZAMANI H., CRESTANI F. & CROFT W. B. (2019). Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, p. 475–484, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3331184.3331265.

BELKIN N. J., COOL C., STEIN A. & THIEL U. (1995). Cases, scripts, and information-seeking strategies : On the design of interactive information retrieval systems. *Expert Systems with Applications*, **9**(3), 379–395. DOI : https ://doi.org/10.1016/0957-4174(95)00011-W.

BOTTOU L. (2010). Large-scale machine learning with stochastic gradient descent. *Proc. of COMPSTAT*. DOI : 10.1007/978-3-7908-2604-3_16.

CAMPOS J. A., CHO K., OTEGI A., SOROA A., AGIRRE E. & AZKUNE G. (2020a). Improving conversational question answering systems after deployment using feedback-weighted learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 2561–2571, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.230.

CAMPOS J. A., OTEGI A., SOROA A., DERIU J., CIELIEBAK M. & AGIRRE E. (2020b). DoQA - accessing domain-specific FAQs via conversational QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7302–7314, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.652.

CHEN T. & GUESTRIN C. (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 785–794, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/2939672.2939785.

CHRISTMANN P., SAHA ROY R., ABUJABAL A., SINGH J. & WEIKUM G. (2019). Look before you hop : Conversational question answering over knowledge graphs using judicious context expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, p. 729–738, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3357384.3358016.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

GAO J., GALLEY M. & LI L. (2019). Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, **13**(2-3), 127–298. DOI : 10.1561/1500000074.

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.

HOI S. C. H., SAHOO D., LU J. & ZHAO P. (2018). Online learning : A comprehensive survey.

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.

LI J., MILLER A. H., CHOPRA S., RANZATO M. & WESTON J. (2016). Dialogue learning with human-in-the-loop. *CoRR*, **abs/1611.09823**.

LIU B., TÜR G., HAKKANI-TÜR D., SHAH P. & HECK L. (2018). Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. 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. 2060–2069, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1187.

PRABOWO D. A. & BUDI HERWANTO G. (2019). Duplicate question detection in question answer website using convolutional neural network. In *2019 5th International Conference on Science and Technology (ICST)*, volume 1, p. 1–6. DOI : 10.1109/ICST47872.2019.9166343.

QU C., YANG L., CHEN C., QIU M., CROFT W. B. & IYYER M. (2020). Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, p. 539–548, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3397271.3401110.

QU C., YANG L., CROFT W. B., ZHANG Y., TRIPPAS J. R. & QIU M. (2019). User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, CHIIR '19, p. 25–33, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3295750.3298924.

RAJPURKAR P., ZHANG J., LOPYREV K. & LIANG P. (2016). SQuAD : 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 2383–2392, Austin, Texas : Association for Computational Linguistics. DOI : 10.18653/v1/D16-1264.

REDDY S., CHEN D. & MANNING C. D. (2019). CoQA : A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, **7**, 249–266. DOI : 10.1162/tacl_a_00266.

TELLEX S., KATZ B., LIN J., FERNANDES A. & MARTON G. (2003). Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International*

ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR '03, p. 41–47, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/860435.860445.

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

Extraction de fragments syntaxiques en français à partir d'une mesure d'autonomie basée sur l'entropie

Marine Courtin[1]
(1) LPP - Laboratoire de Phonétique et Phonologie - UMR 7018 , 19 rue des Bernardins 75005 Paris, France
marine.courtin@sorbonne-nouvelle.fr

RÉSUMÉ

Dans cet article nous nous intéressons à la prédiction du caractère syntaxique ou non d'une séquence de tokens dans des corpus du français. En particulier, nous comparons une méthode d'extraction de fragments syntaxiques identifiés au moyen d'une mesure d'autonomie basée sur l'entropie à une méthode de référence qui extrait des fragments aléatoires. Les résultats semblent indiquer que les fragments ainsi extraits sont bien plus souvent des unités syntaxiques que les fragments aléatoires. Une telle méthode pourrait être utilisée dans des travaux ultérieurs afin de proposer une induction non-supervisée de structures de dépendances syntaxiques.

ABSTRACT

Mining French syntactic fragments using an entropy-based autonomy measure.

In this paper we investigate how sequences of tokens can be identified as syntactic fragments in French corpora. We compare two methods for extracting syntactic fragments : a random baseline and a method that uses an entropy-based autonomy measure to induce syntactic fragments. Results suggest that the proposed method improves the prediction accuracy of sequences that are syntactic units, as compared to the baseline. These findings could be used in further works for unsupervised syntactic dependency induction.

MOTS-CLÉS : fragments syntaxiques, autonomie, analyse syntaxique non-supervisée.

KEYWORDS: syntactic fragments, autonomy, unsupervised parsing.

1 Objectifs

Cet article est une contribution à l'induction non-supervisée de structures syntaxiques. Nous nous plaçons dans le cadre de la syntaxe de dépendance, une théorie syntaxique introduite par (Tesnière, 1959), et cherchons à décider si des segments contigus à l'intérieur d'un énoncé forment des fragments connexes de l'arbre de dépendance. (Gerdes & Kahane, 2011) ont montré que la structure de connexion d'un énoncé pouvait être entièrement définie à partir de l'ensemble des fragments de cet énoncé, fragments qui sont définis notamment par leur capacité à être autonomisés. Nous proposons donc d'utiliser une mesure d'autonomie basée sur l'entropie afin d'induire des fragments syntaxiques. Cette mesure d'autonomie a été utilisée avec succès par le passé pour identifier des unités plus petites telles que des mots, mais nous cherchons à savoir s'il est possible d'extraire des unités plus grandes pour une tâche d'induction de structure syntaxique. Notre hypothèse est que l'évolution de l'entropie à travers la phrase peut nous permettre de faire des prédictions éclairées sur les frontières entre unités syntaxiques, et pourrait se révéler utile pour décider quelles séquences sont des unités syntaxiques et

Volume 2 : 23e REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL), pages 15–27.

quelles séquences ne le sont pas.

Ce travail s'inscrit dans la lignée d'autres travaux qui se sont penchés sur des tâches comme l'analyse syntaxique non-supervisée, l'induction de structures syntaxiques ou encore la recherche d'informations syntaxiques dans des représentations vectorielles denses (ou plongements) au moyen de « structural probes » (Hewitt & Manning, 2019).

Computationnellement, la méthode que nous proposons est plus légère que ces derniers, qui nécessitent d'entraîner des modèles plus lourds comme BERT (Devlin *et al.*, 2019) et ELMO (Peters *et al.*, 2018), ce qui requiert de très grand corpus d'entraînement et la mobilisation de lourdes infrastructures de calcul coûteuses en ressources (Strubell *et al.*, 2019). De plus, nous espérons que l'autonomie basée sur l'entropie soit plus facilement interprétable, puisqu'elle est associée à un solide ancrage théorique en linguistique notamment avec l'hypothèse de Harris (Harris, 1955) qui propose qu'un plus grand paradigme de successeurs ou prédecesseurs à une position entre deux tokens (dans son cas des caractères), indique la présence d'une frontière linguistique (pour lui des frontières entre morphèmes). Cette théorie nous semble assez naturellement adaptable aux frontières entre unités syntaxiques, même si nous pouvons nous demander si l'entropie constituera une information suffisante dans ce cas, puisque la variabilité des tokens est bien plus grande que la variabilité des caractères.

D'autres travaux encore cherchent à établir des liens entre des prédicteurs et la présence de relations de dépendance, c'est le cas par exemple de (Futrell *et al.*, 2019) qui remarquent un lien entre l'information mutuelle pour une paire de mots, et la présence d'une relation de dépendance entre eux. L'information mutuelle étant liée à l'entropie, il nous paraît d'autant plus intéressant d'utiliser une mesure d'autonomie qui soit basée sur cette dernière.

Nous commencerons par présenter en section 2 la mesure d'autonomie utilisée pour prédire la nature syntaxique ou non d'une unité. En section 3 nous présenterons le corpus qui nous servira à entraîner le modèle d'estimation de l'autonomie, ainsi que les corpus arborés sur lesquels seront évaluées nos prédictions. En 4 nous décrirons brièvement le processus d'extraction des fragments aléatoires qui nous servira de méthode de référence. Enfin, en 5 nous présenterons nos premiers résultats.

2 Autonomie et unités syntaxiques

2.1 Mesure d'autonomie

La mesure d'autonomie que nous utilisons est décrite dans (Magistry, 2013). Elle consiste à considérer comme autonome une unité dont les éléments seraient cohésifs, et dont les frontières seraient difficilement prédictibles car situées à des positions de forte entropie.

La mesure d'autonomie est construite de la façon suivante : tout d'abord **l'entropie de branchement** est évaluée à chaque position inter-mot. Cette entropie de branchement permet de rendre compte de la diversité des tokens qui peuvent succéder ou précéder un certain contexte. On calcule ensuite **la variation d'entropie de branchement** qui est obtenue en soustrayant l'entropie de branchement de la position précédente à l'entropie de branchement de la position actuelle. Cette mesure permet d'observer à quel point l'entropie augmente ou diminue en ajoutant un nouveau token.

L'autonomie d'un n-gramme est ensuite calculée en sommant les variations de l'entropie de branche-

ment [1] depuis un parcours gauche-droite et un parcours droite-gauche du texte. Plus un n-gramme aura une autonomie élevée, plus ses frontières auront de fortes entropies comparées aux positions inter-mots, et plus il sera probable que le n-gramme constitue une unité syntaxique.

Formellement, le calcul pour arriver à cette autonomie est réalisé ainsi (nous reprenons toujours (Magistry, 2013)) :

Étant donné un n-gramme $x_{0..n} = x_{0..1}x_{1..2}...x_{n-1..n}$ avec pour contexte gauche $X_{\rightarrow}$, l'entropie de branchement droite est définie comme :

$$h_{\rightarrow}(x_{0..n}) = H(X_{\rightarrow}|x_{0..n})$$

$$h_{\rightarrow}(x_{0..n}) = -\sum_{x \in X_{\rightarrow}} P(x|x_{0..n}) log P(x|x_{0..n}).$$

Pour l'entropie de branchement gauche on note $X_{\leftarrow}$ le contexte droit de $x_{0..n}$, ce qui nous donne :

$$h_{\leftarrow}(x_{0..n}) = H(X_{\leftarrow}|x_{0..n})$$

.

À partir de l'entropie de branchement pour les n-grammes $x_{0..n}$ et $x_{0..n-1}$ la variation d'entropie dans les deux directions peut être calculée :

$$\delta h_{\rightarrow}(x_{0..n}) = h_{\rightarrow}(x_{0..n}) - h_{\rightarrow}(x_{0..n-1})$$

$$\delta h_{\leftarrow}(x_{0..n}) = h_{\leftarrow}(x_{0..n}) - h_{\leftarrow}(x_{1..n})$$

Après avoir appliqué la normalisation mentionnée dans la note 1, l'autonomie du n-gramme $x_{0..n}$ est formée :

$$a(x_{0..n}) = \widetilde{\delta} h_{\leftarrow}(x_{0..n}) + \widetilde{\delta} h_{\rightarrow}(x_{0..n})$$

Cette méthode assignant une autonomie à chaque n-gramme, il est ensuite possible de calculer le score d'une segmentation en additionnant pour chaque n-gramme le produit de son autonomie et de sa taille (en terme de tokens). On a donc une méthode qui nous permet de classer les différentes segmentations d'après leur score global, et un score d'autonomie pour chaque n-gramme.

2.2 Unités syntaxiques

Cette mesure d'autonomie a été pensée pour identifier des mots, mais nous pensons qu'il est possible d'utiliser la même logique pour identifier d'autres unités plus grandes, qui seraient de nature syntaxique. Plus précisément, nous cherchons à identifier des séquences de tokens qui forment une partie connexe dans la structure de dépendance, c'est-à-dire des catenas (Osborne *et al.*, 2012).

1. Une normalisation est également effectuée pour centrer la mesure sur 0 pour chaque taille de n-gramme, afin que les n-grammes plus courts ne soient pas favorisés.

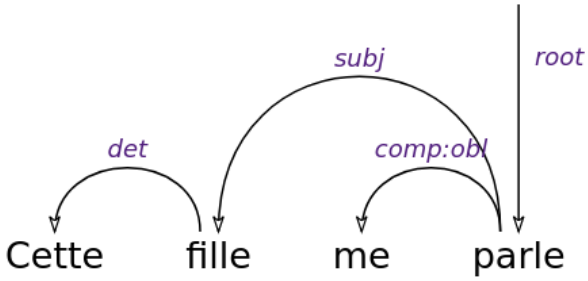


FIGURE 1 – Arbre de dépendance pour l'énoncé « Cette fille me parle »

Ainsi, si nous prenons pour exemple l'énoncé « Cette fille me parle » dont la structure de dépendance peut être représentée comme en figure 1, nous pouvons identifier 10 catenas : (Cette), (fille), (me), (parle), (Cette fille), (fille parle), (me parle), (Cette fille parle), (fille me parle), (Cette fille me parle). En revanche les séquences (fille me) et (Cette fille me) ne sont pas des catenas, puisqu'elles ne constituent pas une partie connexe de la structure de dépendance.

Ce sont donc ces portions connexes de la structure de dépendance, un type d'unités syntaxiques appelées catena, que nous allons chercher à extraire par la suite.

3 Données et méthodologie

3.1 Données

Nous utilisons deux corpus du français : un corpus brut qui est utilisé afin d'entraîner le modèle d'autonomie, et un corpus arboré en dépendance, sur lequel nous poursuivrons l'entraînement du modèle d'autonomie (en utilisant uniquement le texte). Inclure le texte des corpus arborés nous permet d'assurer que le vocabulaire qui y apparaît sera bien couvert. En ce qui concerne les structures de dépendances des corpus arborés, nous les utilisons uniquement afin d'évaluer les prédictions d'unités syntaxiques en comparant les unités prédites avec la structure de référence.

Le premier de ces corpus est constitué d'oeuvres littéraires et segmenté en phrases et en tokens. Nous échantillons des sous-corpus de tailles variées afin d'étudier l'influence de la taille du corpus d'entraînement sur les prédictions. En ce qui concerne les corpus arborés, nous utilisons 6 corpus provenant du projet Universal Dependencies (Zeman *et al.*, 2020), dans la version 2.7 : FQB, GSD, ParTUT, PUD, Sequoia and Spoken. Au total, ces corpus sont constitués de 26 555 phrases et 50 9257 tokens. Ils forment un corpus hétérogène en terme de modalité et de genre, puisqu'on y retrouve de l'écrit et de l'oral, et que les genres couvrent articles de presse, notices de médicaments, wikis, blogs, textes légaux et oral transcrit.

À l'intérieur de ces corpus arborés, des noeuds qui ne correspondent pas directement à des tokens ont été introduits pour rendre compte des amalgames comme « au » à+le, ou « du » de+le. Puisque ces formes désamalgamés n'apparaîtront pas dans notre corpus d'entraînement qui est un corpus brut, nous choisissons d'appliquer une grammaire de réécriture sur ces corpus arborés en utilisant

Grew (Guillaume *et al.*, 2012), afin de rétablir les tokens d'origine en fusionnant les amalgames. [2] Un exemple de cette transformation est présenté en figure 2.

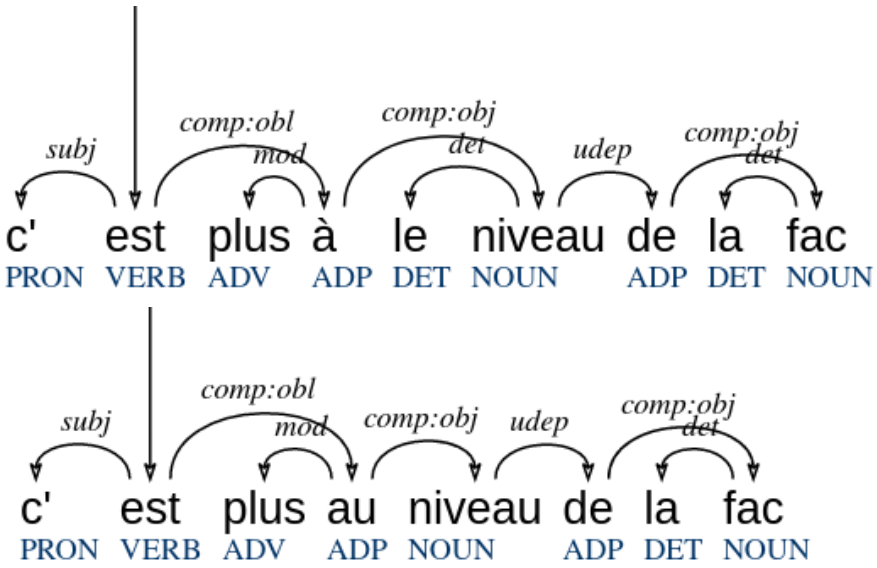


FIGURE 2 – Exemple de transformation pour fusionner un amalgame (à+le → au)

Un autre aspect qui nous semble très intéressant est l'influence du schéma d'annotation sur l'évaluation de notre méthode. Selon le schéma sélectionné, les séquences qui vont être considérées comme des unités syntaxiques vont varier, ce qui signifie que les performances du modèle seront conditionnées par ce schéma. Par exemple, il est possible qu'un fragment extrait par le modèle soit une catena dans un schéma avec des têtes fonctionnelles, mais ne le soit pas dans un schéma avec des têtes lexicales. Afin de tester à quel point ce critère est important, nous évaluons nos prédictions sur 4 versions différentes des corpus arborés, obtenues après application de grammaires de réécriture des graphes de dépendance. Les différences entre ces 4 versions peuvent être décrites de la manière suivante :

- version UD : le schéma d'annotation original pour tous les corpus arborés à l'exception de GSD et Spoken qui sont maintenus en version SUD. Dans cette version les têtes sont des éléments lexicaux et les mots fonctions sont dépendants, ce qui créé des structures généralement plus plates. Une description plus complète des différences entre schéma UD et schéma SUD peut être trouvée dans (Gerdes *et al.*, 2018).
- version SUD : le schéma d'annotation natif pour les corpus GSD et Spoken, contrairement au schéma UD, les têtes sont fonctionnelles, ce qui mène généralement à des structures plus profondes.
- version SUD+ : une version plus extrême du schéma SUD, qui lui est identique en tout aspect à l'exception des relations entre noms et déterminants qui sont inversées pour que les déterminants deviennent têtes. Les autres relations restent identiques.
- version SUD++ : une version identique à la précédente, avec en plus les anciens dépendants du nom qui sont rattachés au déterminant, pour que celui-ci domine tous les éléments à l'intérieur d'un groupe nominal.

Nous savons que ces choix sur le schéma d'annotation vont modifier de façon plus ou moins importante les structures rencontrées, ce qui aura un impact sur l'évaluation du modèle. En première observation

2. La grammaire correspondante réalisée par Bruno Guillaume est disponible ici : https://github.com/surfacesyntacticud/tools/blob/master/textform_wordform/remove_amalg_fr.grs

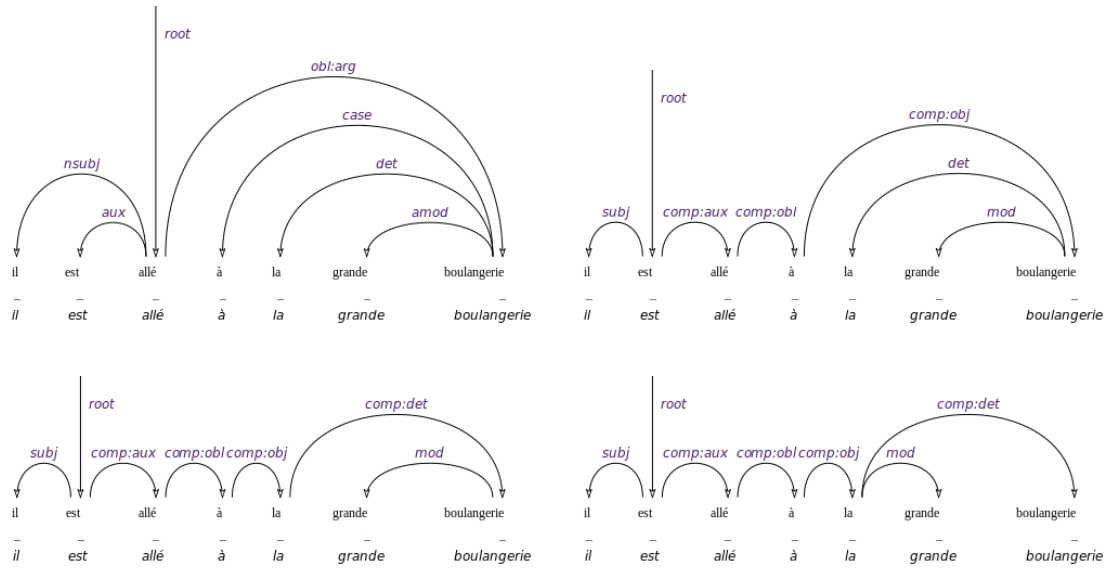


FIGURE 3 – Exemple d'annotation pour les 4 schémas (de gauche à droite puis de haut en bas : UD, SUD, SUD+, SUD++)

nous calculons la proportion de bigrammes, trigrammes et quadrigrammes qui sont des catenas dans les différentes versions des corpus arborés. Plus ces proportions seront élevées, plus il sera probable que le modèle en extrait un nombre important, sans que cela signifie nécessairement qu'il s'améliore. Les résultats dans le tableau 1 nous permettent d'observer que la version SUD+ est de loin la plus riche en catenas observées sur les bigrammes, trigrammes et quadrigrammes, et que la version UD est celle qui en présente le moins, les versions SUD et SUD++ étant similaires en proportions et se positionnant entre la version UD et la version SUD+.

Schéma	Bigrammes	Trigrammes	Quadrigrammes	Tous
UD	0,47	0,39	0,33	0,40
SUD	0,62	0,53	0,46	0,54
SUD+	0,72	0,60	0,51	0,61
SUD++	0,63	0,52	0,46	0,54

TABLE 1 – Proportion des séquences de longueur 2 à 4 qui sont des catenas dans les différents schémas d'annotation des corpus arborés.

3.2 Méthodologie

Dans la section 2 nous avons décrit la mesure d'autonomie que nous utilisons pour extraire des fragments que nous espérons être des unités syntaxiques. Cette mesure est implémentée dans l'outil ELeVE [3] (Magistry & Sagot, 2012) que nous utilisons pour obtenir les fragments.

L'outil nous permet de calculer l'autonomie pour tous les n-grammes, mais aussi d'ordonner les segmentations selon leur score global et éventuellement d'en extraire les n meilleures. Ces informations sont particulièrement intéressantes car le score d'autonomie d'une séquence ne dépend

3. L'outil est disponible en ligne ici : https://github.com/kodexlab/eleve

pas du contexte dans lequel on trouve celle-ci, puisqu'il est calculé à partir de tous ses contextes. En revanche, pour savoir si la séquence constitue bien une unité syntaxique, on aimerait que ce contexte d'apparition soit pris en compte, ce qui est le cas lorsqu'on s'intéresse au score global d'une segmentation. Ainsi une segmentation dans laquelle un seul des segments obtient un score très élevé et tous les autres segments ont des scores médiocres apparaîtra plus bas dans le classement qu'une segmentation qui permet d'obtenir plusieurs segments avec de bons scores, même si individuellement chacun de ces scores est plus faible que celui du très bon segment de la première segmentation.

Ainsi, pour chaque phrase, nous extrayons une liste de fragments, chacun associé à un unique score d'autonomie, et au rang des différentes segmentations dans lesquelles il apparaît. Nous fixons également la taille maximale des n-grammes à comptabiliser à 5 (les estimations pour des segments de longueur supérieures ne seraient pas assez fiables), ce qui nous donnera des fragments de longueur 1 à 4.

En ce qui concerne l'évaluation nous nous intéressons à deux aspects. Le premier consiste à regarder si les fragments sélectionnés constituent bien des catenas dans l'arbre de dépendance du corpus de référence. La proportion de ces fragments sélectionnés qui sont des catenas nous fournira notre score de précision. Nous mesurons aussi à quel point la structure de dépendance est couverte par les fragments extraits, c'est-à-dire quelle proportion des catenas présentes dans la structure nous avons réussi à extraire, ce qui constituera notre rappel.

4 Fragments aléatoires

Nous proposons d'induire aléatoirement des fragments afin de comparer notre méthode à une référence. Si notre hypothèse est vérifiée, nous devrions observer une meilleure compatibilité des fragments induits en utilisant la mesure d'autonomie par rapport aux fragments induits aléatoirement.

Tout d'abord, il nous semble important de préciser la différence entre deux procédés aléatoires permettant d'échantillonner des séquences de tokens :

Une **segmentation aléatoire** consiste à proposer un découpage unique de la phrase, le plus souvent en la parcourant et en attribuant une probabilité d'introduire une frontière à chaque position inter-mot.

Par opposition, une **fragmentation aléatoire** vise à induire plusieurs segmentations, ce qui permettra notamment d'avoir des fragments qui se chevauchent. C'est cette deuxième option que nous privilégions puisque sa sortie ressemblera davantage aux fragments induits.

Parmi les nombreuses façons possibles et imaginables de proposer une fragmentation aléatoire, nous proposons la suivante :

Chaque token dans la phrase est considéré comme noyau d'un fragment aléatoire. Pour ce fragment nous tirons au sort une longueur entre 2 et 4 (puisque ce sont les longueurs que nos fragments candidats peuvent adopter). Une fois la longueur du fragment définie, nous tirons au sort la position du token à l'intérieur du fragment (premier, second, troisième, quatrième). Si la position est incompatible avec la position du token dans la phrase, nous réiterons jusqu'à obtenir une position compatible.

Par exemple pour la phrase « Nous tirons une position au sort », nous pourrions avoir ce type de proposition pour « tirons » :

— longueur du fragment : 3

— position à l'intérieur du fragment : troisième (impossible), premier (possible)

Ce qui nous donnerait un fragment aléatoire « tirons une position ».

Cette première fragmentation aléatoire est appelée « uniforme », puisqu'il n'y a pas de pondération particulière sur la longueur des fragments, celles-ci étant équiprobables.

Nous proposons également une seconde version, appelée « pondérée », avec une pondération sur les longueurs de fragments, afin que la distribution des longueurs dans la fragmentation originale et aléatoire soient similaires. Les poids sélectionnés sont les suivants : 0,77 pour les fragments de longueur 2, 0,15 pour les fragments de longueur 3 et 0,08 pour les fragments de longueur 4.

5 Résultats et discussion

Dans cette partie, nous présentons et analysons des premiers résultats issus des expériences menées, et montrons que l'autonomie pourrait nous permettre d'extraire des fragments syntaxiques.

5.1 Taille du corpus d'entraînement

Afin d'obtenir de bonnes estimations de l'entropie sur les bigrammes, trigrammes et quadrigrammes, il nous faut un corpus d'entraînement suffisamment grand. Nous commençons par regarder la précision sur les fragments extraits pour différentes tailles de corpus : 1000 tokens, 10 000 tokens, 100 000 tokens, 500 000 tokens et 1 million de tokens.

Nous extrayons les fragments apparaissant dans la meilleure segmentation de chaque phrase, et vérifions leur statut de catena dans la version SUD des corpus arborés. Les résultats correspondant sont présentés en 4 où on note principalement que les meilleures précisions globales (respectivement 0,83 et 0,82) sont obtenues pour les deux plus grandes tailles de corpus. Ce gain de précision vient avant tout de meilleures prédictions sur les trigrammes et les quadrigrammes qui sont probablement trop rares dans les petits corpus pour qu'on puisse réellement estimer leur autonomie.

5.2 Influence du schéma d'annotation

Cette fois-ci, nous nous intéressons aux variations dans l'évaluation de la précision selon le schéma d'annotation des corpus arborés.

Les scores de précision globaux indiquent que les fragments extraits respectent davantage le schéma SUD+ (0,81), que le schéma SUD (0,68), c'est-à-dire qu'en changeant uniquement la relation entre les noms et déterminants pour que les déterminants deviennent gouverneurs nous gagnons 0,13 de précision ce qui est considérable. Il est aussi intéressant de noter qu'il y a au final peu de différences entre les scores pour le schéma SUD et le schéma UD, bien que les structures dans ces deux versions soient très différentes.

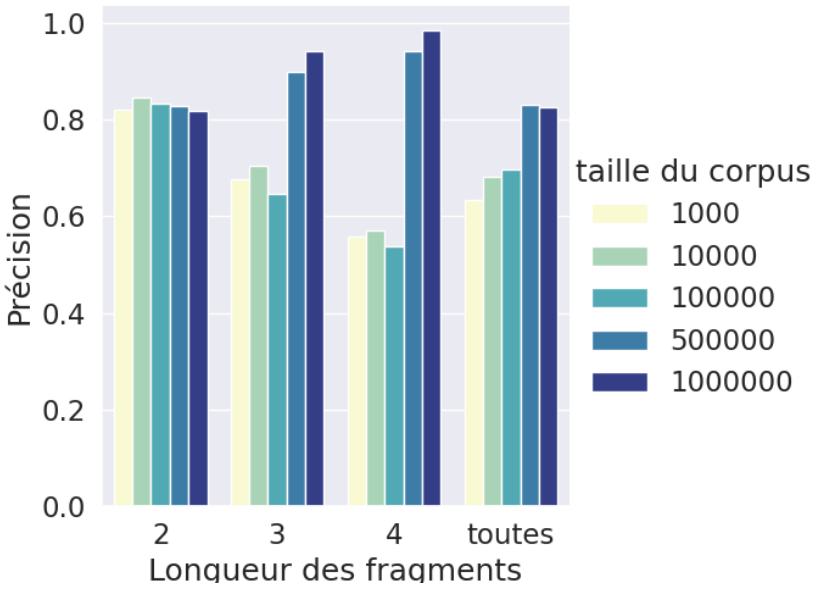


FIGURE 4 – Influence de la taille du corpus d'entraînement sur la précision des fragments extraits (schéma : SUD, n=1)

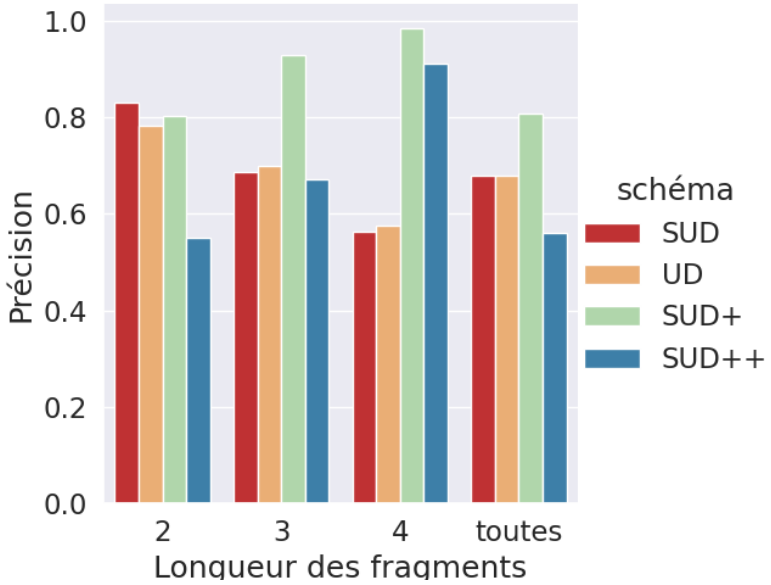


FIGURE 5 – Influence du schéma d'annotation sur la précision des fragments extraits (taille : 1 million de tokens, n=1

5.3 Évolution des scores en fonction des *n* meilleurs

Jusqu'à présent l'évaluation ne concernait que les fragments appartenant à la meilleure segmentation de chaque phrase. Pour ces fragments nous notons une bonne amélioration par rapport à au score de référence sur les fragments aléatoires, en revanche il serait incomplet de s'arrêter ici sans parler de rappel.

Afin de visualiser l'évolution de la précision et du rappel en fonction des *n* meilleures segmentations sélectionnées, nous choisissons de nous intéresser à des phrases de longueur fixe, ce qui nous permettra de fixer un *n* maximum qui corresponde au nombre total de segmentations possibles. [4] Nous sélectionnons toutes les phrases de longueur 10 et faisons varier *n* entre 1 et 401 afin de couvrir toutes les segmentations possibles.

La précision démarre assez haute avec 86% des fragments extraits qui sont des catenas, puis décroit rapidement dans les 25 meilleures segmentations. Elle décroit ensuite plus lentement jusqu'à atteindre 0,57 lorsque toutes les segmentations sont prises en compte. Côté rappel on observe 3 phases, une augmentation très forte dans les 25 premières segmentations, où l'on atteint 0,48, puis une augmentation forte jusqu'aux alentours de la 250e segmentation (0,96) et une augmentation beaucoup plus lente sur la fin.

Pour pouvoir fixer un *n* qui permettrait d'avoir à la fois une bonne précision et un rappel suffisant, il faudrait regarder à quel point certaines catenas peuvent être déduites d'autres catenas qui se combineraient ensemble (par exemple une catena de longueur 2 qui se combinerait avec une catena de longueur 3, avec l'un des noeuds en commun pourrait permettre de déduire la catena de longueur 4 qui englobe les deux).

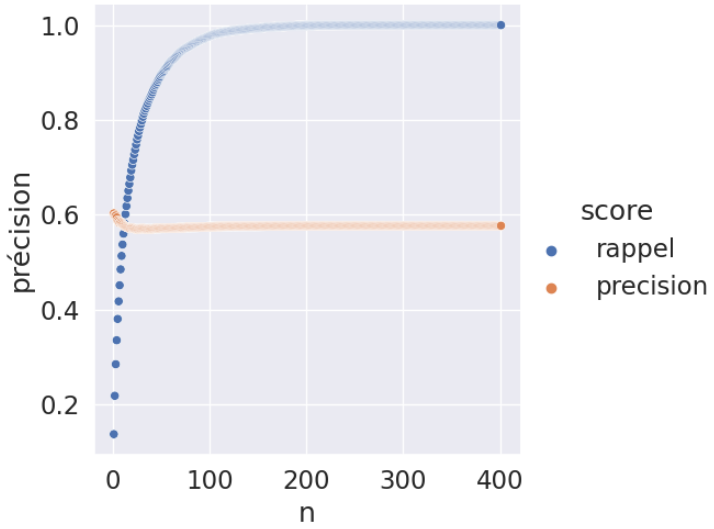


FIGURE 6 – Précision et rappel sur les fragments extraits en fonction des *n* meilleures segmentations sélectionnées (schéma : SUD, taille : 1 million de tokens)

4. Le nombre de segmentations possible pour une phrase de longueur *m* avec des segments de longueurs comprises entre 1 et *p* peut être obtenu à partir de la p-suite de Fibonacci (Olaiju & Taiwo, 2015)

5.4 Comparaison entre fragments extraits et fragments aléatoires

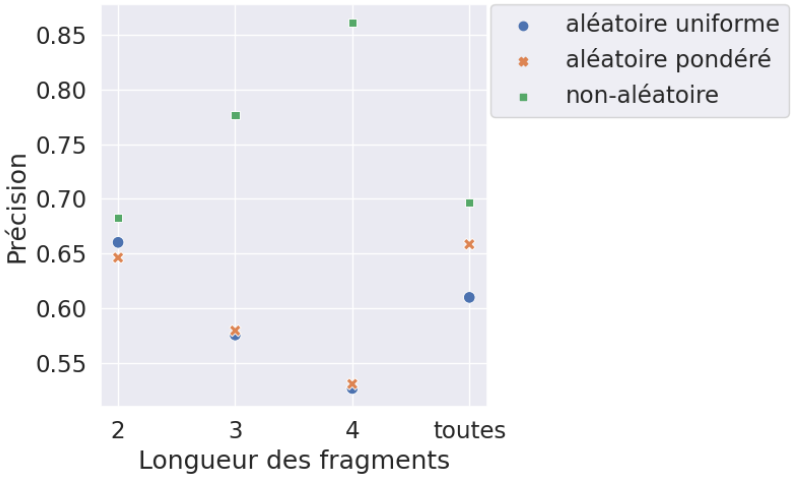


FIGURE 7 – Précision sur les fragments aléatoires et candidats extraits (parmi les 10 meilleures segmentations), en fonction de leur longueur (schéma : SUD, taille : 1 million de tokens)

Dans la figure 7 nous pouvons observer à quel point les fragments extraits (qu'ils soient extraits aléatoirement ou suivant notre méthode) sont effectivement des catenas dans le corpus arboré de référence. Pour ce qui est des fragments aléatoires, nous avons des précisions similaires pour les deux méthodes, avec respectivement pour la méthode uniforme et la méthode pondérée : une précision de 0,66 et 0,65 pour les fragments de longueur 2, 0,58 pour les fragments de longueur 3, 0,53 pour les fragments de longueur 4 et 0,61 contre 0,66 si on ne tient pas compte de la longueur. Plus le fragment est long moins celui-ci a de chances d'être effectivement une catena, ce qui correspond bien aux fréquences décrites dans la table 1.

Pour ce qui est des fragments extraits en suivant notre méthode, on a des scores globalement plus élevés, à savoir 0,68 pour les fragments de longueur 2, 0,78 pour les fragments de longueur 3, 0,86 pour les fragments de longueur 4 et 0,70 si on ne tient pas compte de la longueur. Il est particulièrement intéressant de voir que contrairement aux fragments aléatoires, la précision augmente ici avec la longueur du fragment. Nous pensons que c'est un signe encourageant, car ces catenas sont importantes si on veut pouvoir espérer induire une structure de dépendance, du fait de leur chevauchement avec les autres catenas.

La performance de notre modèle dépend fortement de quel n nous choisissons ici, plus le n sera élevé plus les prédictions seront bruitées et se rapprocheront de la méthode aléatoire qui nous sert de référence. En revanche avec un petit n, on aura de bien meilleures prédictions qu'avec l'aléatoire, mais au détriment du rappel.

6 Conclusion et perspectives

Avec cet article, nous avons voulu montrer qu'il est possible de faire de prédictions sur la nature syntaxique ou non d'une séquence de tokens en français, en nous basant sur l'entropie.

Nous proposons d'extraire des fragments en utilisant une mesure d'autonomie basée sur l'entropie, et montrons que ceux-ci sont plus souvent des unités syntaxiques (plus précisément des catenas) qu'avec une méthode de référence aléatoire. Nous montrons également que le corpus d'entraînement doit atteindre une certaine taille pour pouvoir espérer extraire de plus longs fragments.

Les expériences sur le français semblent indiquer que la structure induite de cette façon se rapproche davantage du schéma SUD+ avec des têtes fonctionnelles et des déterminants têtes des noms avec lesquels ils se combinent, puisque c'est celui-ci qui obtient la meilleure précision.

Il reste encore de nombreuses pistes à explorer, notamment pour savoir quelle serait la couverture minimale permettant d'induire de bonnes structures à partir d'un nombre limité d'unités identifiées, ce qui nous permettrait de sélectionner seulement une partie des meilleurs fragments et d'éviter d'introduire des fragments trop bruités.

Une autre piste consisterait à s'intéresser aux séquences qui semblent les moins probables d'être des unités syntaxiques. Identifier ces « mauvaises » unités pourrait nous permettre d'éliminer d'office un certain nombre de connexions, ce qui réduirait la complexité du problème d'induction de la structure.

Une telle méthode pourrait être utilisée dans des travaux ultérieurs afin de proposer une induction non-supervisée de structures de dépendance syntaxiques.

Références

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.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

FUTRELL R., QIAN P., GIBSON E., FEDORENKO E. & BLANK I. (2019). Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, p. 3–13, Paris, France : Association for Computational Linguistics. DOI : 10.18653/v1/W19-7703.

GERDES K., GUILLAUME B., KAHANE S. & PERRIER G. (2018). SUD or surface-syntactic Universal Dependencies : An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, p. 66–74, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-6008.

GERDES K. & KAHANE S. (2011). Defining dependencies (and constituents). In *International Conference on Dependency linguistics (Depling 2011)*, p. 17–27.

GUILLAUME B., BONFANTE G., MASSON P., MOREY M. & PERRIER G. (2012). Grew : un outil de réécriture de graphes pour le TAL (Grew : a graph rewriting tool for NLP) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 5 : Software Demonstrations*, p. 1–2, Grenoble, France : ATALA/AFCP.

HARRIS Z. S. (1955). From morpheme to phoneme. *Language*, **31**(2), 190–222.

HEWITT J. & MANNING C. D. (2019). A structural probe for finding syntax in word representations. 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. 4129–4138, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1419.

MAGISTRY P. (2013). *Unsupervised Word Segmentation and Wordhood Assessment*. Thèse de doctorat, Paris Diderot ; Inria.

MAGISTRY P. & SAGOT B. (2012). Unsupervized word segmentation : the case for Mandarin Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 383–387, Jeju Island, Korea : Association for Computational Linguistics.

OLAIJU S. & TAIWO A. (2015). Steps problem : the link between combinatoric and k-bonacci sequences. *European Journal of Statistics and Probability*, **3**(4), 10–19.

OSBORNE T., PUTNAM M. & GROSS T. (2012). Catenae : Introducing a novel unit of syntactic analysis. *Syntax*, **15**, 354–396. DOI : https ://doi.org/10.1111/j.1467-9612.2012.00172.x.

PETERS M., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. 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. 2227–2237, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1202.

STRUBELL E., GANESH A. & MCCALLUM A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3645–3650, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1355.

TESNIÈRE L. (1959). *Éléments de syntaxe structurale*.

ZEMAN D., NIVRE J. *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

Les lettres et la machine : un état de l'art en traduction littéraire automatique

Damien Hansen
Université de Liège, CIRTI, 4000 Liège, Belgique
Univ. Grenoble Alpes, CNRS, Grenoble INP*, LIG, 38000 Grenoble, France
damien.hansen@uliege.be

Résumé

Étant donné la récente vague d'intérêt pour la traduction littéraire automatique, cet article vise à recenser les travaux déjà parus sur le sujet, tout en partageant quelques prises de position sur ce thème. Nous commencerons par présenter les travaux précurseurs qui ont motivé ces différentes recherches, ainsi que les résultats obtenus plus récemment dans divers scénarios et pour diverses paires de langues. Pour terminer ce tour d'horizon, nous exposerons les débuts de nos travaux pour la paire anglais-français, avant d'évoquer les préoccupations et les avantages à prendre en compte dans les discussions autour de cette technologie.

Abstract

Machines in the humanities: current state of the art in literary machine translation

Given the recent surge of interest for literary machine translation, this article aims to give an overview of the works already published on the topic, and to share a few points of view on the same subject. First, the pioneering research that motivated these various studies will be presented, as well as the results recently reported in different scenarios and language combinations. To conclude this survey, we will present preliminary findings of our work on the English-French pair, before mentioning some of the concerns and advantages we should include in discussions involving this technology.

Mots-clés : Traduction automatique, post-édition, TAO, littérature, prose, poésie.
Keywords: Machine translation, post-editing, CAT tools, literature, prose, poetry.

1 Introduction

Depuis les premiers travaux en traductologie, la traduction d'ouvrages littéraires a toujours reçu une attention privilégiée et occupé une place un peu à part, en particulier si on les compare à d'autres domaines techniques ou appliqués (Lauvault-Olléon, 2011). Par moment, certains ont voulu qu'elle ne soit pas même possible pour les humains (Genzel *et al.*, 2010 ; Greene *et al.*, 2010 ; Tavailkoski-Shilov, 2019). L'arrivée de la traduction automatique (TA) ne fait pas exception à la règle et reflète d'ailleurs très bien les prises de position diamétralement opposées sur le sujet. D'un côté, il n'est plus surprenant de voir des personnes clamant la parité des systèmes de TA face à la traduction humaine (Hassan *et al.*, 2018), et la littérature est souvent invoquée dans ce cas pour justifier l'observation, même si de nombreuses voix plus mesurées ont depuis remis ce constat en question (Läubli *et al.*, 2018 ; Toral *et al.*, 2018). À l'inverse, ses détracteurs et détractrices ont fréquemment recours aux systèmes en ligne pour montrer que quelques lignes de prose mal traduites suffisent à démontrer sa parfaite inutilité (Toral & Way, 2015b).

* Institute of Engineering Univ. Grenoble Alpes

Pourtant, très peu d'attention a été portée à la traduction littéraire automatique (TLA) jusqu'à présent (Voigt & Jurafsky, 2012 ; Besacier & Schwartz, 2015 ; Moorkens *et al.*, 2018 ; Matusov, 2019). Malgré l'intérêt croissant pour la TA au cours de ces dernières années et les diatribes passionnées qu'elle suscite parfois lorsqu'on la rapproche du secteur littéraire, force est de constater que peu d'études avaient abordé la chose d'un point de vue empirique jusqu'à récemment (Toral & Way, 2015a). Au départ, cet intérêt était plutôt lié à la rencontre de la linguistique computationnelle et de la littérature, comme en atteste le *workshop on computational linguistics for literature*, organisé depuis 2012 par l'ACL. Avant encore, les approches précédant la TA statistique étaient détournées à des fins créatives et appréciées pour la consonance avant-gardiste qu'elles donnaient aux textes produits (Kenny & Winters, 2020).

Aujourd'hui, on constate néanmoins une avancée progressive du modèle de production combinant la TA et post-édition (PE), et ce, depuis les systèmes de TA probabilistes (Besacier & Schwartz, 2015 ; Toral & Way, 2015a). Au vu de ces avancées, il est dès lors devenu légitime de se demander quelle pouvait être la performance de ces outils sur les textes littéraires, d'autant plus que le changement de paradigme apporté par les systèmes neuronaux est réputé donner de meilleurs résultats sur les textes d'une plus grande richesse lexicale et d'une plus grande complexité syntaxique (Bentivogli *et al.*, 2016). Cette question est rendue plus pertinente encore par le fait que la traduction littéraire est et a toujours été indispensable aux échanges culturels à travers le monde, mais aussi par le fait qu'il s'agit d'une tâche particulièrement longue et coûteuse (Voigt & Jurafsky, 2012). Selon Besacier & Schwartz (2015), la TLA pourrait même avoir un intérêt pour chacun des maillons de la chaîne de traduction : des éditeur·trice·s aux lecteur·trice·s, sans oublier les auteur·e·s et les traducteur·trice·s. Dans tous les cas, une revue des travaux déjà accomplis est nécessaire si l'on veut aborder la chose de manière raisonnée.

2 Les précurseurs

Comme nous le remarquions, la traduction a historiquement porté un grand intérêt pour la littérature, et notamment pour la poésie. Ceci est dû en partie à l'idée tenace d'une activité qui serait plus « noble » que les autres et qui lui vaut aujourd'hui d'être décrite par certains, selon Toral & Way (2015a), comme « le dernier bastion de la traduction humaine », renforçant davantage l'idée d'une opposition nette entre l'homme et la machine.

Cette idée n'avait pourtant pas empêché quelques auteur·e·s de s'essayer à la traduction automatique de poésie. Dans l'un des premiers travaux sur la question, Genzel *et al.* (2010) avaient par exemple mis au point un système de TA statistique obéissant à des contraintes de longueur, de métrique accentuelle et de rime. Entraîné sur des données hors domaine, celui-ci parvenait raisonnablement à satisfaire les demandes formelles, au détriment malgré tout de la qualité. Greene *et al.* (2010) avaient pour leur part produit deux systèmes statistiques pareillement contraints par la métrique, qui avaient pour objectif de générer et de traduire des poèmes. Dans le cas du système de traduction, entraîné sur des données à la fois similaires et extérieures au domaine pour augmenter la couverture lexicale, les chercheur·euse·s avaient remarqué qu'il produisait parfois des traductions semblables à un segment de référence ou à des fragments de plusieurs références, parfois de nouvelles traductions valides, et qu'il pouvait être aisément modulé du point de vue de la forme, bien que le résultat n'était pas toujours fluide.

Ces études s'inscrivaient par ailleurs généralement dans une tradition visant à recourir aux capacités de la machine pour mieux comprendre les caractéristiques des textes littéraires. C'est le cas par exemple de Voigt & Jurafsky (2012), dont l'intérêt portait justement sur les difficultés que posent les textes en prose et que doit surmonter la TA (tout comme les humains) pour traduire ce type de textes. Les auteurs relevaient à cet égard l'importance de la cohésion textuelle pour les textes littéraires, en montrant que ceux-ci affichent des chaînes référentielles bien plus denses que d'autres domaines tels que la presse.

Partant, les auteurs suggéraient que les recherches futures sur la TA devraient non seulement pouvoir compter sur des données d'entraînement adaptées au domaine, mais aussi prendre en compte les éléments textuels dépassant le cadre de la phrase, pour que celle-ci soit véritablement efficace en littérature.

Plus récemment, les tentatives de traduction automatique de poèmes ont trouvé des échos grâce aux progrès apportés par la traduction automatique neuronale (TAN). Ghazvininejad *et al.* (2018) ont ainsi présenté un système entraîné sur des données hors domaine et adapté par la suite sur un corpus parallèle de chansons. Les résultats reportés montrent que le modèle parvient à produire des poèmes du français vers l'anglais en respectant les contraintes formelles imposées, et que des humains jugent la production acceptable dans environ 78 % des cas, voire bonne et très bonne pour presque 50 % des exemples. La traduction de poèmes représente toutefois une minorité des recherches sur la traduction littéraire automatique. La plupart d'entre elles se penchent en effet sur la prose, et c'est donc ce versant de la TLA qui nous occupera dans la suite de l'article.

3 Les premières approches statistiques

Comparant justement la traduction automatique de prose et de poésie depuis le français vers l'anglais, Jones & Irvine (2013) offrent une des premières études sur le sujet. Bien que les chercheuses insistent sur l'importance d'évaluer la production d'un point de vue qualitatif, elles offrent tout de même un score BLEU[1] pour la poésie (16.62) et pour la prose (30.05). Dans l'analyse, il ressort que la TA effectue des choix intéressants, mais que les erreurs les plus importantes (choix des temps et choix lexicaux) soulèvent la nécessité d'entraîner ces systèmes sur des données littéraires. La littérature étant en effet un domaine de spécialité, les auteures précisent que cet impératif d'adaptation au domaine vaut donc tout autant que dans d'autres cas d'utilisation de la TA, même si l'objectif des textes créatifs est fondamentalement différent comme nous le verrons plus bas.

Or, on peut trouver des essais d'adaptation pour la prose chez Besacier & Schwartz (2015) et Toral & Way (2015a). Dans le premier cas, une nouvelle est traduite de l'anglais vers le français à l'aide du système Moses (Koehn *et al.*, 2007). Celui-ci est entraîné sur 25 M de phrases hors domaine issues de la campagne IWSLT (Federico *et al.*, 2012) et post-édité ensuite par tranches d'un tiers. Chaque tranche résultante est alors utilisée pour adapter progressivement le système, qui obtient finalement un score BLEU situé aux alentours de 34.79 (par comparaison avec le texte final post-édité). S'il semble que les données ne sont pas suffisantes dans cette étude pour mener à une amélioration du résultat, les auteurs notent en revanche une diminution du temps de PE au fil de l'exercice, bien qu'il soit difficile de savoir si elle est due à l'adaptation. La durée totale de la tâche est tout de même fortement réduite par rapport à une traduction libre (divisée par deux avec une PE non experte), ce qui aurait l'avantage de réduire le coût de la traduction. Les chercheurs évoquent néanmoins la nécessité de s'assurer que la qualité n'y est pas sacrifiée, car la TA a tendance à copier le texte source et ne tient pas compte des références culturelles.

Le second cas est plus proche de Voigt & Jurafsky (2012), dans le sens où les auteurs partagent un intérêt pour les caractéristiques textuelles des textes originaux et de ceux traduits par la machine. Les mesures choisies ici sont deux propriétés qui jouent typiquement en défaveur de la TA lors de la traduction d'ouvrages littéraires ou d'articles de presse, par comparaison avec d'autres domaines (technique et institutionnel par exemple). Premièrement, le degré de spécialisation du domaine (*domain narrowness*) est évalué en fonction du caractère prédictif des modèles de langues entraînés en espagnol pour chaque domaine. Étonnamment, le corpus de presse termine loin devant le corpus littéraire, qui arrive au centre, plus proche des documents techniques et institutionnels. Dans le deuxième cas, une comparaison du degré de liberté calculé pour les traductions anglais-espagnol et espagnol-catalan montre que le domaine importe peu sur la prise de liberté, tandis que la proximité entre les langues paraît plus que déterminante.

[1] BLEU (Papineni *et al.*, 2002) est la métrique la plus utilisée pour comparer les systèmes de TA.

Toujours dans Toral & Way (2015a), les auteurs évaluent la traduction d'un roman de l'espagnol vers le catalan, qu'ils comparent à celle d'un quotidien pour le même couple de langues. Le système utilisé est entraîné principalement à partir d'un corpus d'articles de presse (et de données issues du Web pour le modèle de langue), auxquels sont ajoutés deux romans d'un même auteur pour créer un second système adapté au domaine littéraire. Dans ce cas, il apparaît que l'adaptation au domaine permet d'améliorer sensiblement les résultats (jusqu'à 53.76 points BLEU, contre 49.15 pour le système maison entraîné uniquement sur le corpus de presse et 46.52 pour Google Traduction). Les auteurs indiquent également que 20 % des phrases produites par le meilleur système sont en tous points identiques à la référence, bien qu'il s'agisse des phrases les plus courtes (7,15 % du total de mots), et que 10 % de phrases supplémentaires ne diffèrent de la référence que par cinq caractères ou moins. Par ailleurs, deux locuteur·trice·s bilingues chargé·e·s d'évaluer une sélection de 101 segments originaux et leur traduction rapportent avoir trouvé environ 40 % des phrases de la TA équivalentes à la référence, et presque 20 % de meilleure qualité (en réalité à cause des prises de liberté du traducteur), pour un total de 60 % des phrases affichant une qualité au moins égale à la traduction humaine.

4 L'approche neuronale

Ces trois études pionnières pointent ainsi toutes les trois le besoin de spécialiser les systèmes sur des données littéraires. Les deux essais présentés pour la traduction automatique probabiliste, cependant, n'ont que peu de données à leur disposition, ce qui s'explique en outre par la faible quantité de corpus disponibles dans ce domaine. L'une des conclusions tirées par Besacier & Schwartz (2015) est qu'il faudrait précisément pouvoir compter sur une plus grande représentativité des textes littéraires. Or, nous pouvons constater aujourd'hui un changement en la matière. Celui-ci a été accompagné par la venue de la traduction neuronale, et en particulier des réseaux de neurones récurrents avec mécanisme d'attention (Bahdanau *et al.*, 2015), pour lesquels l'adaptation au domaine (*domain adaptation*) est justement devenue un sujet de choix, y compris en littérature. Dans ce contexte, nous partageons ici quelques publications parues depuis 2018, en particulier autour du projet démarré dans Toral & Way (2018), qui est probablement le plus conséquent à ce jour. Remarquons aussi que l'année 2019 aura été marquée par l'apparition simultanée d'études pour plusieurs autres paires de langue, notamment grâce à l'organisation du premier atelier consacré aux « Qualités de la traduction littéraire automatique ».

4.1 La paire anglais-slovène

L'une de ces publications concerne la paire anglais-slovène, pour laquelle Kuzman *et al.* (2019) mettent au point plusieurs systèmes de traduction personnalisés dont les performances sont comparées à celles de Google Traduction. Ces systèmes sont entraînés via le module OpenNMT (Klein *et al.*, 2017) sur des données génériques (OPUS[2]), auxquelles sont ajoutés soit un corpus de textes littéraires variés (9 romans du corpus SPOOK[3]) soit un roman de la même auteure et de la même traductrice que le texte à évaluer. Dans le premier cas, le score BLEU obtenu est de 18.50, contre 19.01 et 20.75 pour le second. Les chercheur·euse·s reportent par ailleurs des scores pour d'autres systèmes entraînés uniquement sur le roman (1.78), uniquement sur le corpus littéraire varié (6.61) ainsi que sur un mélange de toutes les données (16.02). Étonnamment, l'ajout de données littéraires supplémentaires semble faire baisser le score, par comparaison au système entraîné sur le roman et sur le corpus hors domaine. Google Traduction en revanche affiche des résultats supérieurs dans tous les cas (21.97), ce que les auteur·e·s attribuent à la taille fortement limitée de leur corpus d'entraînement, et ce, bien que l'ajout de données adaptées à une auteure et à sa traductrice permette de se rapprocher de ce score.

[2] Cf. Tiedemann (2012).
[3] Cf. http://nl.ijs.si/spook/.

4.2 La paire gaélique écossais-irlandais

Dans la foulée, Ó Murchú (2019) s'est essayé lui aussi à la post-édition d'un texte littéraire, à traduire du gaélique écossais vers l'irlandais. Intergaelic, l'outil utilisé pour cette tâche, est un système hybride disponible en ligne et entraîné sur un corpus relativement restreint. Sans surprise, le texte ainsi produit comporte à la fois des phrases parfaites en l'état et des passages qui devaient entièrement être retraduits, bien que le résultat soit largement correct d'un point de vue grammatical. Par comparaison avec la traduction humaine produite par le même auteur, la TA atteint un score BLEU de 35, mais paraît moins naturelle par endroit et plus proche de la structure du texte source. De plus, le traducteur note que certains éléments ne sont pas traduits et qu'une intervention humaine est presque systématiquement requise dans le cas des noms propres, des régionalismes, des néologismes et des interjections. Il ajoute cependant que la PE est 31 % plus rapide et qu'elle lui a permis d'éviter des erreurs. Les réactions des maisons d'édition contactées et de l'auteur original vis-à-vis de ce projet ont par ailleurs été très positives et le roman sera quant à lui publié.

4.3 La paire anglais-russe

Matusov (2019) s'est lui aussi intéressé à la TLA, en tentant pour sa part d'évaluer un système pour la traduction de l'anglais vers de russe et un second de l'allemand vers l'anglais. Les systèmes en question sont entraînés grâce au module RETURNN (Zeyer *et al.*, 2018). Dans le premier scénario, les corpus utilisés sont un mélange de données hors domaine et de données littéraires (270 K phrases provenant d'OPUS Books[4]), augmentées avec un corpus synthétique de romans (2.3 M de phrases[5]) à la manière de Sennrich *et al.* (2016). Pour ce couple de langues, l'auteur reporte des points BLEU de 13.9 pour Google Traduction, de 14.2 pour le système non adapté et de 15.2 pour le système adapté. Étant donné la faible fiabilité de cette métrique (comme nous le verrons ci-dessous), une évaluation humaine a également été conduite, sous la forme d'une classification d'erreur personnalisée. Dans l'ensemble, l'évaluation indique que 20 % des phrases ne contiennent pas d'erreur et que le système maison donne lieu à moins d'erreurs graves (de sens et de syntaxe) que Google Traduction, bien qu'il produise plus d'erreurs mineures. Les erreurs de sens demeurent toutefois les plus fréquentes, tandis que le nombre d'omissions, les erreurs liées à la cohérence et les problèmes de registre sont jugés trop nombreux pour un extrait de cette taille.

4.4 La paire allemand-anglais

Dans le même article, Matusov (2019) met au point un second système pour la paire allemand-anglais. Contrairement aux études précédentes, celui-ci est entraîné sur l'architecture Transformer (Vaswani *et al.*, 2017), qui a plus récemment permis d'atteindre de meilleurs résultats dans de nombreux domaines. Les corpus utilisés pour ce faire sont les mêmes (50 K phrases d'OPUS Books pour les textes littéraires), à l'exception cependant de celui utilisé pour la rétrotraduction du corpus synthétique (10 M de phrases issues du projet Gutenberg[6]). Cette fois, les résultats sont à l'inverse du cas précédent, avec un score BLEU de 20.2 pour Google Traduction, de 18.5 pour le système généraliste et de 16.2 pour le système adapté. Cette baisse pourrait néanmoins être expliquée par la taille réduite des données d'entraînement dans ce second scénario, ou tout simplement par la rigidité de la mesure BLEU. L'auteur indique d'ailleurs que ces résultats sont à prendre avec un peu de recul, car l'utilisation de la métrique TER montre des résultats tout à fait inverses, ce que constataient déjà Kuzman *et al.* (2019) dans certains cas.

4 Cf. Tiedemann (2012).
5 Issues du site Web Lib.ru.
6 Cf. http://www.gutenberg.org/about/.

Ceci se note également dans l'évaluation humaine, puisque les juges estiment de meilleure qualité les textes produits par le système ayant pourtant obtenu le score le plus faible. En effet, 30 % des phrases sont jugées être d'une qualité acceptable. Il apparaît d'autre part que le système de TA adapté utilise un vocabulaire beaucoup plus riche, bien que les erreurs de sens soient plus fréquentes par comparaison avec Google Traduction. Malgré la difficulté liée à la fiabilité des métriques, l'auteur conclut donc que l'adaptation au domaine permet d'obtenir de meilleurs résultats en littérature, en particulier pour le couple allemand-anglais. Cette pratique pourrait en outre faciliter le processus de post-édition, puisque les erreurs les plus graves et les problèmes de syntaxes laissent place à des erreurs plus simples et rapides à corriger. Enfin, il est noté que des systèmes de traduction automatique opérant au niveau du texte et non plus de la phrase pourraient grandement bénéficier à la prise en charge des textes littéraires.

4.5 La paire anglais-néerlandais

Cet intérêt pour les types d'erreurs produits par la TA se note également chez d'autres chercheur·euse·s, intéressé·e·s pour leur part à la traduction littéraire automatique depuis l'anglais vers le néerlandais (Tezcan *et al.*, 2019 ; Fonteyne *et al.*, 2020). Si aucun système sur-mesure n'est proposé dans ce cas, c'est surtout parce que les auteur·e·s cherchent ici à relever les caractéristiques stylistiques et à classifier les erreurs les plus fréquentes des textes produits par les outils en ligne tels que Google Traduction. Dans ce sens, des mesures de richesse lexicale, d'entropie, de couverture lexicale et sémantique et d'équivalence syntaxique sont utilisées. Ces différents éléments sont ainsi censés refléter des traits esthétiques essentiels des textes littéraires, dans la mesure où l'expérience de lecture serait en partie tributaire de celles-ci et où la machine devrait absolument les préserver pour être considérée véritablement efficace dans ce domaine (Tezcan *et al.*, 2019). Concernant la richesse lexicale, il apparaît que le texte produit par la TA dépasse non seulement le texte source, mais aussi la référence humaine. D'autres mesures confirment cette tendance, en montrant cependant un écart plus réduit. Cette richesse apparente pourrait aussi être due en partie aux traductions erronées, ce que semble confirmer la mesure d'entropie en présentant la traduction humaine comme étant plus imprévisible. L'analyse de recouvrement lexical indique quant à elle une faible différence entre les deux traductions, ce qui pourrait signifier que la TA possède un degré de cohésion moins élevé, mais cet écart est en revanche beaucoup plus marqué pour l'équivalence syntaxique, confirmant à nouveau que la TA reste plus proche de la structure du texte source.

Pour ce qui est de l'annotation d'erreurs, les chercheur·euse·s ont employé la taxonomie SCATE (Tezcan *et al.*, 2017), évaluant les deux aspects typiques de fluidité (grammaire, vocabulaire, orthographe…) et de fidélité (ajout, omission, glissement de sens…), et ont étendu celle-ci avec deux critères supplémentaires de fluidité concernant le style et le registre. Pour ces mesures, les résultats indiquent que 44 % des phrases ne contiendraient aucune erreur et que la TAN, connue pour ses difficultés à traiter de longues phrases, parviendrait à ne pas commettre d'erreur dans des phrases allant jusqu'à un maximum de 37 mots. Comme dans d'autres études pour cette paire de langues, les erreurs de fluidité sont plus nombreuses que les erreurs de fidélité, les erreurs liées à la cohérence représentant plus de 50 % des occurrences dans le premier cas et les traductions erronées plus de 80 % dans le deuxième. Les problèmes de style et de registre arrivent quant à eux en troisième position.

Dans une seconde étude, Fonteyne *et al.* (2020) proposent de répéter cette même expérience sur un extrait plus long du texte source évalué par un second juge, afin de vérifier l'accord inter-annotateur. Après avoir pointé les difficultés inhérentes aux annotations d'erreur et nuancé les scores qui semblaient faibles à première vue, les auteur·e·s confirment ainsi les résultats obtenus précédemment concernant le nombre de phrases sans erreur (44 %), la prévalence des erreurs de fluidité sur les erreurs de fidélité, ainsi que les types d'erreurs les plus fréquents. Ils ajoutent en fin de compte qu'il faudrait davantage

d'études sur le sujet afin de voir notamment quels effets ces résultats peuvent avoir sur l'expérience de lecture, car c'est évidemment sur cet objectif que le domaine littéraire se démarque de tous les autres.

4.6 La paire anglais-catalan

D'autres auteur·e·s se sont récemment penché·e·s sur cette question, au cours d'un projet qui réunit déjà quelques publications (Toral & Way, 2018 ; Toral *et al.*, 2018 ; Moorkens *et al.*, 2018 ; Toral *et al.*, 2020 ; Guerberof-Arenas & Toral, 2020). Celui-ci vise à évaluer les performances de la TA sur la paire anglais-catalan en confrontant différents paradigmes. Il est par ailleurs intéressant de noter que ce projet est le premier à utiliser des données d'entraînement uniquement littéraires, là où les études qui précèdent avaient systématiquement recours à des corpus hors domaine afin d'obtenir des données suffisamment larges pour la mise au point du système de TA. Dans certains cas, les données littéraires sont d'ailleurs presque négligeables, ce qui tient une fois encore à la difficulté de trouver des corpus adaptés.

Selon Toral & Way (2018), néanmoins, l'émergence continue des livres électroniques pourrait changer cette réalité, en facilitant la construction de corpus parallèles. Si l'on y ajoute l'arrivée de la TA neuronale, et en particulier sa capacité à produire des traductions moins littérales et lexicalement plus riches que les systèmes statistiques, ces changements pourraient, à en croire les auteurs, changer notre perception vis-à-vis de l'utilisation de cette technologie en littérature. De fait, le corpus d'entraînement utilisé ici a été constitué de manière automatique à partir de 133 romans (1 M de phrases) et de leur traduction. Plus encore, les auteurs ont eux aussi constitué un corpus synthétique, composé d'environ 1 000 romans (5 M de phrases), pour renforcer la représentation de la langue cible. Le système est entraîné avec le module Nematus (Sennrich *et al.*, 2017), et le modèle résultant est un assemblage des quatre meilleures sorties.

Le système neuronal est ensuite comparé à un système statistique, entraîné lui aussi sur ces données. C'est le module Moses qui a servi ici à mettre au point le modèle, pour lequel un corpus hors domaine de 400 K phrases (OpenSubtitles2018[7]) et un autre corpus synthétique de 16 M de phrases (caWaC[8]) ont également été utilisés. Les deux paradigmes sont alors évalués sur 12 romans différents, parus entre 1920 et aujourd'hui. Les scores BLEU obtenus varient entre 17.94 et 38.92 pour la TAN, avec une moyenne de 32.12, contre 16.11 et 37.35 pour la variante statistique, avec une moyenne de 29.09. La variante neuronale montre donc de meilleurs résultats dans tous les cas, avec une augmentation du score BLEU pouvant varier de 3 à 14 % (10 % en moyenne)[9].

Toujours dans Toral & Way (2018), une évaluation humaine conduite sur trois des romans traduits montre que la TA neuronale est perçue d'une qualité équivalente à la traduction humaine pour 17, 32 et 34 % des cas, contre 7, 18 et 20 % pour la TA statistique. Les juges n'étant pas des traducteurs experts, une seconde évaluation est conduite dans Toral *et al.* (2020) et donne alors 15, 32 et 30 % de phrases produites par la TA équivalentes à la traduction humaine pour les mêmes romans. Une évaluation de l'effort de post-édition indique par ailleurs que l'entraînement des systèmes sur des données littéraires permet de réduire considérablement le nombre d'erreurs. Par comparaison avec un texte issu de Google Traduction, celles-ci se limitent en outre à des cas moins sévères, comme le montrait déjà Matusov (2019). Les comparaisons effectuées dans ce travail montrent en dernier lieu un net saut de performance de l'architecture Transformer par rapport aux modèles récurrents avec attention.

---

7 Cf. Lison & Tiedemann (2016) ; http://www.opensubtitles.org/.

8 Cf. Ljubešić & Toral (2014).

9 Toral *et al.* (2020) proposent des comparaisons avec d'autres systèmes sur ces mêmes données et obtiennent un score BLEU dépassant les 40, bien qu'ils ne donnent pas les chiffres exacts.

Ceci se note également lorsque l'on demande à des professionnel·le·s de post-éditer un texte produit par ces deux systèmes. Dans Toral *et al.* (2018), il a ainsi été demandé à six juges de comparer la traduction libre d'un roman et sa post-édition par les systèmes statistiques et neuronaux mentionnés plus haut[10]. Comme dans d'autres études sur la post-édition, les conclusions indiquent que cette tâche diminue l'effort cognitif fourni durant la traduction (42 %), réduit le nombre de frappes (23 %) et augmente la productivité (36 %), y compris lorsqu'elle mène à des pauses plus longues que la traduction libre (25 %). Toutes ces mesures sont les plus marquées lorsque l'on passe du système statistique au système neuronal. Selon Moorkens *et al.* (2018), les traducteur·trice·s n'ont cependant pas toujours conscience de ces gains, et tou·te·s expriment une préférence pour la traduction sans post-édition, car le sentiment de liberté et de créativité y est plus grand. Les chercheur·euse·s remarquent par ailleurs que la plus jeune génération de traducteur·trice·s montre une attitude plus positive envers la TA et accepteraient plus facilement d'y avoir recours que leurs collègues plus expérimenté·e·s. Enfin, l'unique juge à avoir plus d'un an d'expérience avec la post-édition est également celui qui trouve l'aide offerte par la TA la plus utile.

Dans une autre étude, Guerberof-Arenas & Toral (2020) reportent qu'un traducteur et une traductrice littéraires trouvent la TA plutôt utile et qu'elle produit des textes plutôt fidèles que fluides. À choisir entre la post-édition et la traduction libre, les deux ne s'accordent toutefois pas sur la méthode la plus rapide et la moins laborieuse, bien que la deuxième soit toujours préférée. Le retour d'un juge expert pointe d'autre part le fait que le texte littéraire post-édité contiendrait moins d'erreurs, mais aussi moins de tournures créatives. Enfin, une enquête sur la réception de la traduction humaine et post-éditée chez les lecteur·trice·s montre que les deux paradigmes sont évalués presque au même niveau en ce qui concerne leur réception, leur appréciation et l'intérêt suscité, malgré la difficulté à dégager des tendances. Pour cette dernière raison les auteur·e·s concluent à la nécessité de vérifier ces observations avec plusieurs paires de langues, avec d'autres ouvrages, et avec différents genres littéraires.

5 Réévaluer la paire anglais-français avec l'approche neuronale

Dans l'ensemble, donc, nous voyons que les systèmes neuronaux produisent de meilleurs résultats que les systèmes statistiques, en particulier ceux basés sur Transformer, de même que ceux entraînés sur de la littérature par comparaison aux généralistes, pour autant qu'ils soient suffisamment fournis en données. Chacun de ces changements offre ainsi des gains de performances pour le domaine littéraire, de même qu'un aperçu plus objectif des résultats que l'on pourrait attendre dans ce secteur. Selon Kuzman *et al.* (2019) et Toral *et al.* (2020), il serait également utile de voir quelle serait la performance d'un système de TA entraîné spécifiquement sur la production des mêmes auteur·e·s et traducteur·trice·s. Or, c'est précisément sur cette question que se centre notre projet de recherche, qui offrira par la même occasion de nouveaux résultats pour la paire anglais-français grâce à l'approche neuronale. Nos premiers essais, effectués sur un corpus restreint de 6 tomes (45 K phrases) issus d'une saga d'*heroic fantasy*, donnent un score BLEU de 9.24, tandis que Google Traduction et DeepL obtiennent respectivement 10.79 et 10.04. De cela, nous notons tout d'abord le fait qu'un corpus aussi petit obtienne un score comparable aux systèmes en ligne, mais aussi et surtout le faible score de ces systèmes qui obtiennent généralement des points BLEU se situant entre 20 et 30 pour de grands classiques de la littérature. Ceci, nous l'attribuons aux particularités de notre roman, qui affiche un registre relativement soutenu, des régionalismes, des tournures volontairement vieillies, ainsi que de nombreux concepts et néologismes propres à la série, mieux définis comme des irréalias (Loponen, 2009). Sans surprise, l'ajout de données génériques permet de dépasser ces scores et de se rapprocher de ceux obtenus dans les autres langues. Nos prochains travaux s'attarderont donc sur ce processus d'adaptation au domaine et sur les évaluations, automatiques et humaines, de ce texte visiblement plus complexe que d'autres pour la TA.

[10] Il est important de noter que les participant·e·s n'avaient pas tou·te·s une maîtrise parfaite de la langue source et que deux seulement avaient de l'expérience en post-édition.

6 Inconvénients et préoccupations

Évidemment, le développement de la traduction automatique laisse envisager divers changements et pose de nombreuses questions, certaines plus positives que d'autres. S'il est peut-être encore tôt pour se prononcer sur l'évolution de la TLA, celle-ci soulève tout de même des enjeux éthiques et sociétaux qui concernent la profession au sens large. Certains d'entre eux ont déjà été soulevés dans d'autres domaines, mais pourraient déjà ou prochainement concerner la littérature. Pour cette raison, nous pensons qu'il convient de considérer ceux-ci au plus tôt, pour assurer un avenir éthique et durable de la traduction littéraire (Taivalkoski-Shilov, 2019), et de le faire du point de vue de l'humain (Kenny, 2017).

Tout au long de cet article, nous avons par exemple abordé des problématiques de qualité et de créativité. S'il est difficile de tirer des conclusions sur ce sujet, en raison des résultats souvent contradictoires, il reste toutefois certain qu'une mauvaise implémentation de la TA conduirait inévitablement à une baisse de la qualité des textes produits. Celle-ci dépend en effet intimement des conditions de production et des facteurs sociaux qui entourent l'acte de traduction (Taivalkoski-Shilov, 2019). Or, une baisse de qualité pourrait à son tour avoir des conséquences négatives sur l'expérience de lecture, sur la reconnaissance du métier de traducteur·trice et sur le travail de l'auteur·e, mais aussi sur le transfert de culture, l'apprentissage des langues et les compétences linguistiques des lecteur·trice·s (*Ibid*.). Heureusement, les résultats sur ce sujet nous semblent montrer que ces effets sont dus avant tout à un manque d'expérience en post-édition. Le changement de paradigme introduit par la technologie nécessite bien sûr d'être formé et familiarisé à une tout autre méthode de travail pour pouvoir traduire dans des conditions normales et confortables. Ceci vaut tout autant pour la traduction assistée par ordinateur (TAO), dont l'utilisation en littérature a initialement suscité les mêmes rejets que la TA, mais que l'on voit apparaître de plus en plus fréquemment à mesure que les gens s'habituent à ces outils.

Les effets négatifs d'une interface peu ergonomique sur le travail et sur la qualité de la traduction résultante sont d'ailleurs d'ores et déjà bien documentés pour la TAO (Teixeira & O'Brien, 2017). Si le processus est mal conçu, la TA pourrait donc pareillement augmenter la charge cognitive des utilisateur·trice·s (Taivalkoski-Shilov, 2019) et faire en sorte qu'ils ou elles puissent difficilement se détacher des suggestions pour trouver des solutions plus créatives (Şahin & Gürses, 2019). Toral & Way (2015b) insistent dès lors sur le fait que l'introduction de la TA en littérature devrait se faire d'une manière qui s'éloignerait de la post-édition traditionnelle et qui correspondrait mieux à ce domaine. Pour cette raison, notamment, Besacier & Schwartz (2015) mentionnent l'intérêt d'une interface de PE interactive capable d'afficher plusieurs propositions, tandis que Toral & Way (2015a) évoquent un système susceptible d'être entraîné au fur et à mesure que le traducteur travaille. Enfin, le manque de contexte en PE peut mener à un manque d'homogénéité (Besacier & Schwartz, 2015), c'est pourquoi une segmentation par paragraphe devrait être possible dans l'idéal (Moorkens *et al*., 2018 ; Nunes Vieira *et al.*, 2020), ce qui vaut également pour l'utilisation des logiciels de TAO en littérature selon nous.

Ces deux préoccupations pointent de ce fait la nécessité d'intégrer pleinement les traducteurs et les traductrices au centre de ces recherches, de façon à pouvoir adapter les outils à leur travail (Ruffo, 2018). La connaissance de la technologie et l'amélioration des interfaces de travail ne sont cependant pas les seules choses à prendre en considération, puisque les traducteur·trice·s littéraires dépendent aussi des groupes d'édition. L'objectif de ces derniers restant principalement financier, l'arrivée de la TA pose potentiellement un risque pour leurs conditions de travail (Taivalkoski-Shilov, 2019). Comme l'a montré l'utilisation des outils de TAO et de la TA dans d'autres domaines, ces outils conduisent généralement à une baisse de la rémunération et à des délais encore plus courts. Dans des cas plus graves, certains vendent directement des sorties de TA non post-éditées, et l'on pourrait penser que d'autres éditeur·trice·s soient tenté·e·s d'engager des post-éditeur·trice·s non professionnel·le·s à l'avenir (*Ibid*.).

Dans la même ligne d'idée, Şahin & Gürses (2019) ajoutent que les systèmes de TA soulèvent la question du plagiat. Celle-ci serait particulièrement problématique en Turquie, selon les auteurs, où des éditeurs publient des retraductions en partie plagiées, et cette préoccupation est d'autant plus grande si ces outils sont utilisés sans intervention humaine. Néanmoins, la question des droits et de la propriété sur les données se pose quant à elle dans tous les cas, y compris lorsque des humains sont intégrés dans le processus, bien qu'il n'y ait pas encore de régulation à ce sujet (Taivalkoski-Shilov, 2019). De même, les outils automatiques posent un risque pour la visibilité des traducteurs et traductrices (Cronin, 2013). Ceci vaut bien évidemment pour tous les domaines de la traduction, mais peut-être plus encore dans le monde de l'édition, où le manque de reconnaissance représente déjà un défi de taille. Heureusement, aucun des ouvrages considérés ici n'envisage la TA autrement que comme un outil au service des professionnel·le·s, excepté lorsque ces outils servent d'aide aux apprenant·e·s d'une langue étrangère.

Kenny & Winters (2020b), de leur côté, soulignent le fait que la traduction automatique aurait tendance à atténuer la voix des traducteur·trice·s dans le texte cible. Il a aussi souvent été rapporté que les professionnel·le·s se sentent contraint·e·s par la tâche de post-édition et la segmentation par phrases, ce qui pourrait coïncider avec une baisse de la créativité. Toral *et al.* (2018) remarquent en effet que les traducteur·trice·s prennent très peu de liberté par rapport à la structure du texte source. Nunes Vieira *et al.* (2020), en revanche, notent une totale prise de liberté lorsque ces mêmes traducteurs et traductrices possèdent de l'expérience en post-édition. Dans ce même cas, les juges experts ne notent pas de différence concernant la créativité si l'on compare les textes post-édités aux références humaines (*Ibid.*). D'autre part, la TA pourrait au contraire stimuler la démarche créative en offrant d'autres possibilités de traduction ou en facilitant la traduction des passages moins importants, de manière à ce que l'humain puisse se concentrer sur les points les plus difficiles, les passages clefs où l'ingéniosité humaine est nécessaire (Şahin & Gürses, 2019).

7 Avantages

En plus d'une possible augmentation de la créativité, l'utilisation de la TA dans le domaine littéraire laisse se profiler d'autres points positifs. Selon Besacier & Schwartz (2015), on pourrait d'ailleurs y trouver un intérêt à tous les niveaux de la chaîne de traduction. Pour les traducteur·trice·s, ce sont bien entendu les possibles augmentations de la qualité et de la productivité qui ont été largement évoquées jusqu'ici. Selon Taivalkoski-Shilov (2019), les outils tels que la TA et la TAO pourraient en outre réduire la charge cognitive des utilisateur·trice·s et leur permettre de focaliser leur attention sur les tâches les plus complexes. Dans l'ensemble, ces avantages ont donc le potentiel de rendre le travail des traducteur·trice·s plus agréable (*Ibid.*), en particulier s'ils sont susceptibles d'aboutir à une prise d'importance de la dimension créative, comme nous le pensons. L'enjeu est d'autant plus important que le statut des traducteur·trice·s littéraires est déjà extrêmement précaire et les délais de traduction souvent très courts. Évidemment, cela implique de pouvoir apporter des assurances concernant les enjeux éthiques que nous venons de voir.

Pour les éditeur·trice·s, le principal avantage résiderait sans surprise dans la réduction des coûts, dont nous pourrions espérer qu'il profite aux traducteur·trice·s professionnel·e·s. Par ailleurs, la TA pourrait ainsi leur offrir la possibilité d'augmenter les commandes de traductions, faciliter leur accès à de nouveaux ouvrages et diversifier leur catalogue. Pour les auteur·e·s, l'avantage résiderait avant tout dans le fait de voir son œuvre traduite dans un plus grand nombre de langues (Besacier & Schwartz, 2015). Pour les lecteur·trice·s, enfin, ce serait la chance d'avoir accès plus rapidement aux traductions de leurs auteur·e·s favori·te·s (*Ibid.*). La TA serait plus utile encore à celles et ceux qui n'auraient pas accès à leur contenu préféré dans les langues qu'ils pratiquent. Dans ce contexte, la TA se profile comme une aide précieuse à la lecture et à l'apprentissage des langues (Oliver González *et al.*, 2019 ; Matusov, 2019).

Pour terminer, si une utilisation abusive de la TA pose un risque pour la visibilité des traducteur·trice·s, celle-ci pourrait également donner plus de visibilité à d'autres et assurer une plus grande diversité dans le panorama de la traduction littéraire. Les traducteur·trice·s et les auteur·e·s n'ont en effet pas tou·te·s les mêmes chances de voir leurs ouvrages paraître sur le marché de l'édition (Castro, 2020). Les romans écrits dans des langues à plus faible diffusion, par exemple, ont peu de chance d'attirer l'intérêt des maisons d'édition (*Ibid*. ; Toral & Way, 2015b). Or, la traduction automatique pourrait permettre à ces éditeur·trice·s de se familiariser avec d'autres œuvres ou faciliter l'envoi d'un échantillon en vue d'une publication (Matusov, 2019). De la même manière, la TA pourrait en outre favoriser la diversité de la traduction littéraire en introduisant des travaux produits par des personnes issues de minorités (Castro, 2020). Cette dernière observation ne concerne d'ailleurs pas uniquement les personnes victimes de discrimination, mais aussi simplement des traducteur·trice·s émergent·e·s (Tazelaar, 2020) ou des auteur·e·s moins connu·e·s (Matusov, 2019).

8 Discussion

Il n'est pas rare de voir des gens condamner fermement la traduction automatique et son utilisation en littérature, au motif que les résultats obtenus par les systèmes existants seraient bien trop mauvais et que ceux-ci produiraient des traductions trop littérales, là où la traduction des textes littéraires nécessite d'opérer des choix stylistiques signifiants. Or, l'un des avantages reconnus aujourd'hui de la TA neuronale est sa capacité à traduire plus librement, pour autant que les données d'entraînement soient adaptées. Il faut en effet reconnaître qu'aucun des systèmes de traduction auxquels nous avons (librement) accès à ce jour n'a été prévu pour traduire des textes littéraires. Comme le remarquaient déjà Jones & Irvine (2013) :

> « Tout comme les humains, les systèmes de traduction automatique sont capables de produire des traductions qui peuvent être tantôt littérales, tantôt plus libres, et doivent constamment opérer des choix lors du décodage [pour produire une phrase de sortie]. Dans le cas des systèmes SMT [et NMT], ces choix sont dépendants des observations contenues dans les données d'entraînement et de leur fréquence. Lorsqu'ils sont entraînés sur des jeux de données semblables au corpus de test, il est donc probable qu'ils effectuent des choix plus pertinents. »[11]

Pourtant, il aura fallu attendre jusqu'à très récemment pour que des systèmes soient adaptés au domaine littéraire. Et même s'il n'est pas toujours aisé de se faire une idée du niveau de qualité concrètement obtenu en raison du manque d'accès aux résultats, les conclusions reportées semblent prometteuses, bien qu'il reste encore du chemin à parcourir. Les défis particuliers posés par la littérature reflètent en effet les intérêts d'autres pistes de recherche très récentes en traduction automatique, notamment l'augmentation de données (Fadaee *et al*., 2017), l'adaptation au domaine (Chu & Wang, 2018) ou encore la possibilité de traiter les éléments au niveau textuel (Lopes *et al*., 2020).

Dans tous les cas, tou·te·s les auteur·e·s cité·e·s ici se rejoignent sur le fait que l'ingéniosité humaine restera indispensable pour de longues années encore, peu importe le domaine littéraire, technique, etc. Combinée aux capacités de la machine, la créativité humaine pourrait se voir renforcée, tout comme la qualité du texte produit. Cela implique comme nous l'avons dit de pouvoir compter sur une interface ergonomique qui serait spécifiquement prévue à cet effet. Pour cette raison, il nous semble justement que l'environnement de TAO représenterait le point de départ idéal — et qu'il pourrait amener une solution à d'autres problèmes, comme celui de la propriété sur les données — même s'il reste beaucoup de chemin à faire dans ce sens.

[11] Nous traduisons.

En tout état de cause, et comme le montrent particulièrement les résultats de Kuzman *et al.* (2019), l'utilisation de données adaptées au domaine ne suffit pas et nécessite de disposer de très larges corpus d'entraînement, car le besoin en données des systèmes de TA reste primordial. Toutefois, le recours aux corpus littéraires pour l'entraînement des modèles neuronaux permet effectivement d'améliorer les résultats et même d'atteindre des performances encourageantes dans le cas d'un système entraîné uniquement à partir de données littéraires suffisamment larges (Toral *et al.*, 2020).

En revanche, l'évaluation des textes produits par la machine présente encore, à l'heure actuelle, un obstacle de taille. Si la métrique BLEU (Papineni *et al.*, 2002) est encore largement utilisée en pratique et peut offrir une estimation de la qualité par rapport à une (et une seule) référence, cette mesure doit être prise avec une précaution toute particulière dans le cas de la littérature (Kit & Wong, 2015 ; Toral & Way, 2015b) et plus encore si elle concerne des systèmes entraînés sur des couples de langue, des genres et des auteur·e·s qui peuvent varier fortement d'une étude à l'autre. Pour cette raison, certain·e·s chercheur·euse·s partagent d'autres métriques[12], comme METEOR (Denkowski & Lavie, 2014), TER (Snover *et al.*, 2006) ou HTER (Snover *et al.*, 2009). Toutefois, ces mesures varient également d'une publication à l'autre, elles offrent parfois des résultats tout à fait contradictoires entre elles et cette pratique reste minoritaire, bien que de nouvelles métriques reposant moins sur les structures de surface, comme YiSi (Lo, 2019) et COMET (Rei *et al.*, 2020), laissent entrevoir de meilleures corrélations avec les jugements humains.

De plus, ces métriques n'aident pas nécessairement les lecteur·trice·s à se représenter concrètement les résultats produits par la TA. Une seconde solution consiste donc à effectuer des classifications d'erreur ou des évaluations humaines, mais la majorité des auteur·e·s rapportent dans ces deux cas des résultats hautement variables entre chaque juge (Kuzman *et al.*, 2019 ; Nunes Vieira et al., 2020 ; Guerberof-Arenas & Toral, 2020). À titre d'exemple, certains trouvent que la TA est moins utile aux traducteur·trice·s littéraires débutant·e·s (Şahin & Gürses, 2019), là d'autres trouvent précisément le contraire (Moorkens *et al.*, 2018). Enfin, les évaluations humaines sont rendues d'autant plus compliquées, et leurs conclusions d'autant plus difficiles à interpréter, par le fait que les évaluateurs sont familiers soit avec la technologie, soit avec la traduction littéraire, mais jamais dans les deux domaines (Nunes Vieira *et al.*, 2020). Parfois même, les évaluateur·trice·s ont très peu d'expérience dans ces deux tâches.

Finalement, ces études constituent tout de même des pistes de recherche intéressantes, qui ont de surcroît l'avantage de fournir de précieuses informations concernant les caractéristiques des textes créatifs ainsi que le travail particulier des traducteur·trice·s littéraires. La TA, sans être — heureusement — parfaite, reste quant à elle un outil potentiellement utile. Il n'est toutefois pas question de sombrer dans le déterminisme technologique pour autant, et c'est pourquoi ces démarches doivent être accompagnées de réflexions concernant les changements négatifs et positifs qu'elle pourrait apporter. Les auteur·e·s savent à quel point il peut être difficile d'être traduit·e. Il arrive même fréquemment que les lecteur·trice·s perdent la possibilité de suivre les suites d'une série ou de terminer une saga qu'ils avaient entamée, pour de simples raisons de coûts, de droits et de revenus. La situation des traducteur·trice·s littéraires n'est d'ailleurs pas toujours plus confortable, et des avancées dans le domaine de la TLA pourraient ainsi apporter des changements positifs à ces égards, tout comme elle pourrait contribuer à détériorer des conditions déjà précaires. Cette piste de recherche reste donc une exploration fondamentalement intéressante, qui pourrait contribuer à mieux équiper les traducteurs littéraires, mais elle doit nécessairement s'accompagner pour cela de considérations sociologiques qui indiqueraient comment l'intégrer au mieux (comme pour toutes les avancées liées au *deep learning*).

[12] Cf. Toral & Way (2015a) ; Kuzman *et al.* (2019) ; Matusov (2019) ; Guerberof-Arenas & Toral (2020).

Remerciements

Je tiens à remercier mes encadrant·e·s et évaluateur·trice·s pour leur relecture, leurs retours éclairants et leurs suggestions judicieuses.

Références

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: Conference Track Proceedings*, *ICLR 2015*, *San Diego (CA)*, *États-Unis*, *7–9 mai 2015*. ARXIV : 1409.0473.

BENTIVOGLI L., BISAZZA L., CETTOLO M. & FEDERICO M. (2016). Neural versus Phrase-Based Machine Translation Quality: a Case Study. In J. SU, K. DUH & X. CARRERAS, Éds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, *EMNLP 2016*, *Austin (TX)*, *États-Unis*, *1–5 novembre 2016*, p. 257–267 : ACL. DOI : 10.18653/v1/D16-1025, ARXIV : 1608.04631.

BESACIER L. & SCHWARTZ L. (2015). Automated Translation of a Literary Work: A Pilot Study. In A. FELDMAN, A. KAZANTSEVA, S. SZPAKOWICZ & C. KOOLEN, Éds., *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, *NAACL-HLT 2015*, *Denver (CO)*, *États-Unis*, *4 juin 2015*, p. 114–122 : ACL. DOI : 10.3115/v1/W15-0713, HAL : hal-01147903.

CASTRO O. (2020). Transnational Feminism, Women Writers in Translation, Stateless Cultures/Literatures in Translation. Présentation, *Creative Translation and Technologies Expert Meeting*, *Université de Surrey, Royaume-Uni*, *29 mai 2020*.

CHU C. & WANG R. (2018). A Survey of Domain Adaptation for Neural Machine Translation. In E. M. BENDER, L. DERCZYNSKI & P. ISABELLE, Éds., *Proceedings of the 27th International Conference on Computational Linguistics*, *COLING 2018*, *Santa Fe (NM)*, *États-Unis*, *20–26 août 2018*, p. 1304–1319 : ACL.

CRONIN M. (2013). *Translation in the Digital Age*. Routledge.

DENKOWSKI M. & LAVIE A. (2014). Meteor universal: Language specific translation evaluation for any target language. In O. BOJAR, C. BUCK, C. FEDERMANN, B. HADDOW, P. KOEHN, C. MONZ, M. POST & L. SPECIA, Éds., *Proceedings of the Ninth Workshop on Statistical Machine Translation*, *EACL 2014*, *Baltimore (MD)*, *États-Unis*, *26–27 juin 2014*, p. 376–380 : ACL. DOI : 10.3115/v1/W14-3348.

FADAEE M., BISAZZA A. & MONZ C. (2017). Data Augmentation for Low-Resource Neural Machine Translation. In R. BARZILAY & M.-Y. KAN, Éds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, *ACL 2017*, *Vancouver*, *Canada*, *30 juillet – 4 août 2017*, p. 567–573 : ACL. DOI : 10.18653/v1/P17-2090.

FEDERICO M., STÜKER S., BENTIVOGLI L., PAUL M., CETTOLO M., HERRMANN T., NIEHUES J. & MORETTI G. (2012). The IWSLT 2011 Evaluation Campaign on Automatic Talk Translation. In C. CALZOLARI, K. CHOUKRI, T. DECLERCK, M. U. DOĞAN, B. MÆGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the 8th International Conference on Language Resources and Evaluation*, *LREC'12*, *Istanbul*, *Turquie*, *21–27 mai 2012*, p. 3543–3550 : ELRA.

FONTEYNE M., TEZCAN A. & LIEVEN M. (2020). Literary Machine Translation under the Magnifying Glass: Assessing the Quality of an NMT-Translated Detective Novel on Document Level. In N. CALZOLARI, F. BÉCHET, P. BLACHE, K. CHOUKRI, C. CIERI, T. DELRECK, S. GOGGI, H. ISAHARA, B. MAEGAARD, J. MARIANI, H. MAZO, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the 12th Conference on Language Resources and Evaluation*, *LREC 2020*, *Marseille*, *France*, *11–16 mai 2020*, p. 3790–3798 : ELRA.

GENZEL D., USZKOREIT J. & OCH F. (2010). 'Poetic' Statistical Machine Translation: Rhyme and Meter. In H. LI & L. MÀRQUEZ, Éds., *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, *EMNLP '10*, *Cambridge (MA)*, *États-Unis*, *9–11 octobre 2010*, p. 158–166 : ACL.

GHAZVININEJAD M., CHOI Y. & KNIGHT K. (2018). Neural Poetry Translation. In M. WALKER, H. JI & A. STENT, Éds., *Proceedings of NAACL-HLT 2018*, *NAACL-HLT 2018*, *New Orleans (LO)*, *États-Unis*, *1–6 juin 2018*, p. 67–71 : ACL. DOI : 10.18653/v1/N18-2011.

GREENE E., BODRUMLU T. & KNIGHT K. (2010). Automatic Analysis of Rhythmic Poetry with Applications to Generation and Translation. In H. LI & L. MÀRQUEZ, Éds., *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, *EMNLP '10*, *Cambridge (MA)*, *États-Unis*, *9–11 octobre 2010*, p. 524–533 : ACL.

GUERBEROF-ARENAS A. & TORAL A. (2020). The Impact of Post-Editing and Machine Translation on Creativity and Reading Experience. *Translation Spaces*, 9(2), p. 255–282. DOI : 10.1075/ts.20035.gue, ARXIV : 2101.06125.

HASSAN H., AUE A., CHEN C., CHOWDHARY V., CLARK J., FEDERMANN C., HUANG X., JUNCZYS-DOWMUNT M., LEWIS W., LI M., LIU S., LIU T.-Y., LUO R., MENEZES A., QIN T., SEIDE F., TAN X., TIAN F., WU L., WU S., XIA Y., ZHANG D., ZHANG Z. & ZHOU M. (2018). Achieving Human Parity on Automatic Chinese to English News Translation. *ArXiv*. ARXIV : 1803.05567.

JONES R. & IRVINE A. (2013). The (Un)faithful Machine Translator. In P. LENDVAI & K. ZERVANOU, Éds., *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, *LaTeCH 2013*, *Sofia*, *Bulgarie*, *8 août 2013*, p. 96–101 : ACL.

KENNY D., Éd. (2017). *Human Issues in Translation Technology*. Routledge.

KENNY D. & WINTERS M. (2020). Another Way of Looking at Machine Translation and Literary Translation. Présentation, *Creative Translation and Technologies Expert Meeting*, *Université de Surrey, Royaume-Uni*, *29 mai 2020*.

KENNY D. & WINTERS M. (2020b). Machine translation, ethics and the literary translator's voice. *Translation Spaces*, 9(1), p. 123–149. DOI : 10.1075/ts.00024.ken.

KIT C. & WONG T.-M. (2015). Evaluation in Machine Translation and Computer-Aided Translation. In S-W. CHAN, Éd., *The Routledge Encyclopedia of Translation Technology*, p. 213–236. Routeledge.

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 S. ANANIADOU, Éd., *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Companion Volume, Proceedings of the Demo and Poster Sessions*, *ACL 2007*, *Prague*, *République Tchèque*, *25-27 juin 2007*, p. 177–180 : ACL.

KLEIN G., KIM Y., DENG Y., SENELLART J. & RUSH A. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. In M. BANSAL & H. JI, Éds., *Proceedings of ACL 2017, System Demonstrations*, *Vancouver*, *Canada*, *30 juillet–4 août 2017*, p. 67–72 : ACL. ARXIV : 1701.02810.

KUZMAN T., VINTAR Š. & ARČAN M. (2019). Neural Machine Translation of Literary Texts from English to Slovene. In J. HADLEY, M. POPOVIĆ, H. AFLI & A. WAY, Éds., *Proceedings of the Qualities of Literary Machine Translation*, *Machine Translation Summit XVII*, *Dublin*, *Irelande*, *19 août 2019*, p. 1–9 : EAMT.

LÄUBLI S., SENNRICH R. & VOLK M. (2018). Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In E. RILOFF, D. CHIANG, J. HOCKENMAIER & J. TSUJI, Éds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, *EMNLP 2018*, *Bruxelles*, *Belgique*, *31 octobre – 4 novembre 2018*, p. 4791–4796 : ACL. DOI : 10.18653/v1/D18-1512, ARXIV : arXiv:1808.07048.

LAVAULT-OLLÉON E. (2011). L'ergonomie, nouveau paradigme pour la traductologie. *ILCEA*, 14, p. 1–17. DOI : 10.4000/ilcea.1078.

LISON P. & TIEDEMANN J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In C. CALZOLARI, K. CHOUKRI, T. DECLERCK, S. GOGGI, M. GROBELNIK, B. MÆGAARD, J. MARIANI, MAZO H., A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, *LREC'16*, *Portorož*, *Slovénie*, *23–28 mai 2016*, p. 923–929 : ELRA.

LJUBEŠIĆ N. & TORAL A. (2016). caWaC: A web corpus of Catalan and its application to language modeling and machine translation. In C. CALZOLARI, K. CHOUKRI, T. DECLERCK, H. LOFTSSON, B. MÆGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, *LREC'14*, *Reykjavik*, *Islande*, *26–31 mai 2014*, p. 1728–1732 : ELRA.

LO C. (2019). YiSi - a Unified Semantic MT Quality Evaluation and Estimation Metric for Languages with Different Levels of Available Resources. 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. NEVES, M. POST, M. TURCHI & K. VERSPOOR, Éds., *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, *WMT 2019*, *Florence*, *Italie*, *1–2 août 2019*, p. 507–513 : ACL. DOI : 10.18653/v1/W19-5358.

LOPES A., FARAJIAN M. A., BAWDEN R., ZHANG M. & MARTINS A. (2020). Document-level Neural MT: A Systematic Comparison. In A. MARTINS, H. MONIZ, S. FUMEGA, B. MARTINS, F. BATISTA, L. COHEUR, C. PARRA, I. TRANCOSO, M. TURCHI, A. BISAZZA, J. MOORKENS, A. GUERBEROF, M. NURMINEN, L. MARG, M. L. FORCADA, Éds., *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, *EAMT 2020*, *Lisbonne*, *Portugal*, *3–5 novembre 2020*, p. 225–234 : EAMT. HAL : hal-02900686.

LOPONEN M. (2009). Translating Irrealia – Creating a Semiotic Framework for the Translation of Fictional Cultures. *Chinese Semiotic Studies*, 2(1), p. 165–175. DOI : 10.1515/css-2009-0117.

MATUSOV E. (2019). The Challenges of Using Neural Machine Translation for Literature. In J. HADLEY, M. POPOVIĆ, H. AFLI & A. WAY, Éds., *Proceedings of the Qualities of Literary Machine Translation*, *Machine Translation Summit XVII*, *Dublin*, *Irelande*, *19 août 2019*, p. 10–19 : EAMT.

MOORKENS J., TORAL A., CASTILHO S. & WAY A. (2018). Translators' Perceptions of Literary Post-Editing using Statistical and Neural Machine Translation. *Translation Spaces*, 7(2), p. 240–262. DOI : 10.1075/ts.18014.moo.

NUNES VIEIRA L., ZHANG X., YOUDALE R. & CARL M. (2020). Machine Translation and Literary Texts: A Network of Possibilities. Présentation, *Creative Translation and Technologies Expert Meeting*, *Université de Surrey, Royaume-Uni*, *29 mai 2020*.

OLIVER GONZÁLEZ A., TORAL A. & GUERBEROF-ARENAS A. (2019). InLéctor: Neural Machine Translation for the creation of bilingual ebooks. In J. HADLEY, M. POPOVIĆ, H. AFLI & A. WAY, Éds., *Proceedings of the Qualities of Literary Machine Translation*, *Machine Translation Summit XVII*, *Dublin*, *Irelande*, *19 août 2019*, p. VII : EAMT.

Ó MURCHÚ E. P. (2019). Using Intergaelic to pre-translate and subsequently post-edit a sci-fi novel from Scottish Gaelic to Irish. In J. HADLEY, M. POPOVIĆ, H. AFLI & A. WAY, Éds., *Proceedings of the Qualities of Literary Machine Translation*, *Machine Translation Summit XVII*, *Dublin*, *Irelande*, *19 août 2019*, p. 20–25 : EAMT.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2016). Bleu: a Method for Automatic Evaluation of Machine Translation. In P. ISABELLE, E. CHARNIAK & D. LIN, Éds., *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, *ACL-02*, *Philadelphia (PA)*, *États-Unis*, *7–12 juillet 2016*, p. 311–318 : ACL. DOI : 10.3115/1073083.1073135.

REI R., STEWART C., FARINHA A. C. & LAVIE A. (2020). COMET: A Neural Framework for MT Evaluation. In B. WEBBER, T. COHN, Y. HE & Y. LIU, Éds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, *EMNLP 2020*, *16–20 novembre 2020*, p. 2685–2702 : ACL. DOI : 10.18653/v1/2020.emnlp-main.213, ARXIV : 2009.09025.

RUFFO P. (2018). Human-Computer Interaction in Translation: Literary Translators on Technology and Their Roles. In D. CHAMBERS, J. DRUGAN, J. ESTEVES-FERREIRA, J. M. MACAN, R. MITKOV & O.-M. STEFANOV, Éds., *Proceedings of Translating and the Computer 40*, *TC40*, *Londres*, *Royaume-Uni*, *15–16 novembre 2018*, p. 127–131 : Éditions Tradulex.

ŞAHIN M. & GÜRSES S. (2019). Would MT kill creativity in literary retranslation? In J. HADLEY, M. POPOVIĆ, H. AFLI & A. WAY, Éds., *Proceedings of the Qualities of Literary Machine Translation*, *Machine Translation Summit XVII*, *Dublin*, *Irelande*, *19 août 2019*, p. 26–34 : EAMT.

SENNRICH R., HADDOW B. & BIRCH A. (2016). Improving Neural Machine Translation Models with Monolingual Data. In K. ERK & N. SZPAKOWICZ, Éds., *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *ACL 2016*, *Berlin*, *Allemagne*, *7–12 août 2016*, p. 86–96 : ACL. DOI : 10.18653/v1/P16-1009, ARXIV : 1511.06709.

SENNRICH R., FIRAT O., CHO K., BIRCH A., HADDOW B., HITSCHLER J., JUNCZYS-DOWMUNT M., LÄUBLI S., MICELI-BARONE A. V., MOKRY J. & NĂDEJDE M. (2017). Nematus: a Toolkit for Neural Machine Translation. In A. MARTINS & A. PEÑAS, Éds., *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, *EACL 2017*, *Valence*, *Espagne*, *3–7 avril 2017*, p. 65–68 : ACL.

SNOVER M., DORR B., SCHWARTZ R., MICCIULLA L. & WEISCHEDEL R. (2006). A Study of Translation Error Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation of the Americas, AMTA 2006*, *Cambridge (MA)*, *États-Unis*, *8–12 août 2006*, p. 223–231 : ACL.

SNOVER M., MADNANI N., DORR B. & SCHWARTZ R. (2009). Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In C. CALLISON-BURCH, P. KOEHN, C. MONZ & J. SCHROEDER, Éds., *Proceedings of the Fourth Workshop on Statistical Machine Translation, EACL 2009*, *Athènes*, *Grèce*, *30–31 mars 2009*, p. 259–268 : ACL.

TAIVALKOSKI-SHILOV K. (2019). Ethical issues regarding machine(-assisted) translation of literary texts. *Perspectives*, 27(5), p. 689–703. DOI : 10.1080/0907676X.2018.1520907.

TAZELAAR F. (2020). CELA: Connecting Emerging Literary Artists. Présentation, *Creative Translation and Technologies Expert Meeting*, *Université de Surrey, Royaume-Uni*, *29 mai 2020*.

TEIXEIRA C. S. C. & O'BRIEN S. (2017). Investigating the cognitive ergonomic aspects of translation tools in a workplace setting. *Translation Spaces*, 6(1), p. 79–103. DOI : 10.1075/ts.6.1.05tei.

TEZCAN A., HOSTE V. & MACKEN L. (2017). SCATE taxonomy and corpus of machine translation errors. In G. CORPAS PASTOR & I. DURÁN-MUÑOZ, Éds., *Trends in e-tools and resources for translators and interpreters*, p. 219–244. Brill - Rodopi.

TEZCAN A., DAEMS J. & MACKEN L. (2019). When a 'Sport' Is a Person and Other Issues for NMT of Novels. In J. HADLEY, M. POPOVIĆ, H. AFLI & A. WAY, Éds., *Proceedings of the Qualities of Literary Machine Translation*, *Machine Translation Summit XVII*, *Dublin*, *Irelande*, *19 août 2019*, p. 40–19 : EAMT.

TIEDEMANN J. (2012). Parallel Data, Tools and Interfaces in OPUS. In C. CALZOLARI, K. CHOUKRI, T. DECLERCK, M. U. DOĞAN, B. MÆGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the 8th International Conference on Language Resources and Evaluation*, *LREC'12*, *Istanbul*, *Turquie*, *21–27 mai 2012*, p. 2214–2218 : ELRA.

TORAL A., CASTILHO S., HU K. & WAY A. (2018). Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In O. BOJAR, R. CHATTERJEE, C. FEDERMANN, M. FISHEL, Y. GRAHAM, B. HADDOW, M. HUCK, A. JIMENO YEPES, P. KOEHN, C. MONZ, M. NEGRI, A. NÉVÉOL, M. NEVES, M. POST, L. SPECIA, M. TURCHI, K. VERSPOOR, Éds., *Proceedings of the Third Conference on Machine Translation: Research Papers*, *WMT 2018*, *Bruxelles*, *Belgique*, *31 octobre – 1 novembre 2018*, p. 113–123 : ACL. DOI : 10.18653/v1/W18-6312, ARXIV : arXiv:1808.10432.

TORAL A., OLIVER A. & RIBAS-BELLESTÍN P. (2020). Machine Translation of Novels in the Age of Transformer. In J. PORSIEL, Éd., *Maschinelle Übersetzung für Übersetzungsprofis*, p. 276–295. BDÜ Fachverlag. ARXIV : 2011.14979.

TORAL A. & WAY A. (2015a). Translating Literary Text between Related Languages using SMT. In A. FELDMAN, A. KAZANTSEVA, S. SZPAKOWICZ & C. KOOLEN, Éds., *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, *NAACL-HLT 2015*, *Denver (CO)*, *États-Unis*, *4 juin 2015*, p. 123–132 : ACL. DOI : 10.3115/v1/W15-0714.

TORAL A. & WAY A. (2015b). Machine-Assisted Translation of Literary Text: A Case Study. *Translation Spaces*, 4(2), p. 241–268. DOI : 10.1075/ts.4.2.04tor.

TORAL A. & WAY A. (2018). What Level of Quality can Neural Machine Translation Attain on Literary Text? In S. CASTILHO, F. GASPARI & S. DOHERTY, Éds., *Translation Quality Assessment: From Principles to Practice*, p. 263–287. Springer. ARXIV : 1801.04962.

TORAL A., WIELING M. & WAY A. (2018). Post-Editing Effort of a Novel With Statistical and Neural Machine Translation. *Frontiers in Digital Humanities*, 5(9), p. 1–11. DOI : 10.3389/fdigh.2018.00009.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A., KAISER L. & 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*, *NIPS 2017*, *Long Beach (CA)*, *États-Unis*, *4–9 décembre 2017*, p. 5998–6008 : Curran Associates Inc.

VOIGT R. & JURAFSKY D. (2012). Towards a Literary Machine Translation: The Role of Referential Cohesion. In D. ELSON, A. KAZANTSEVA, R. MIHALCEA & S. SZPAKOWICZ, Éds., *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, *NAACL-HLT 2012*, *Montréal*, *Canada*, *8 juin 2012*, p. 18–25 : ACL.

ZEYER A., ALKHOULI T. & NEY H. (2018). RETURNN as a generic flexible neural toolkit with application to translation and speech recognition. In F. LIU & T. SOLORIO, Éds., *Proceedings of ACL 2018, System Demonstrations*, *Melbourne*, *Australie*, *15–20 juillet 2018*, p. 128–133 : ACL. DOI : 10.18653/v1/P18-4022, ARXIV : 1805.05225.

Deuxième partie

Posters

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

Adaptation de ressources en langue anglaise pour interroger des données tabulaires en français

Alexis Blandin[1, 2] (1) IRISA, EXPRESSION, Vannes, France
(2) UNEEK, 44000 Nantes, France
alexis.blandin@univ-ubs.fr

RÉSUMÉ

Les récents développements des approches d'apprentissage neuronal profond ont permis des avancées très significatives dans le domaine de l'interrogation des systèmes d'information en langage naturel. Cependant, pour le français, les ressources à disposition ne permettent de considérer que les requêtes sur des données stockées sous forme de texte. Or, aujourd'hui la majorité des données utilisées en entreprise sont stockées sous forme tabulaire. Il est donc intéressant d'évaluer si les ressources anglophones associées (jeux de données tabulaires et modèles) peuvent être adaptées au français tout en conservant de bons résultats.

ABSTRACT

Adaptation of resources in English to query French tabular data

Research in artificial intelligence has led to significant progress in the processing of questions in French natural language through machine learning. However, for the French language, the resources available only allow us to consider queries on data stored in text form. But, available data used in companies repositories are stored in tabular form, it is therefore of particular interest to evaluate whether the available English-language resources (tabular data and models) can be effectively adapted to the French language while maintaining a good information retrieval performance.

MOTS-CLÉS : Traitement du langage naturel, recherche d'information, intelligence artificielle.

KEYWORDS: Natural language processing, information retrieval, artificial intelligence.

1 Introduction

Ces dernières années les avancées en traitement automatique du langage permettent d'envisager des applications plus poussées dans des milieux professionnels, comme par exemple l'amélioration du traitement des données pour la gestion de la relation client (CRM). Ainsi le traitement des questions en langage naturel sur des données peut être très prometteur.

Le traitement des questions en langage naturel a connu un vif intérêt, à l'image du jeu de données SQuAD présenté par (Rajpurkar *et al.*, 2018), qui fait désormais partie du jeu de tâche du benchmark GLUE (Wang *et al.*, 2019). La tâche consiste à extraire les portions de textes qui permettent de répondre à la question posée. Or, la plupart des données sont stockées sous forme d'une base de données tabulaire. Ainsi, dans le cadre de données stockées dans une base SQL, le contexte des questions serait une table de données, et la réponse serait alors une traduction de cette question en SQL.

Plusieurs jeux de données ont été réalisés afin de répondre à cette tâche d'analyse de données tabulaires, que ce soit en se focalisant sur des questions générales et complexes (Pasupat & Liang, 2015), plusieurs questions simples (Iyyer *et al.*, 2017), ou en cherchant une traduction la plus fidèle qui soit d'un langage naturel vers le SQL (Zhong *et al.*, 2017).
Si chacun de ces jeux de données propose un modèle associé, on peut toutefois remarquer que le modèle TAPAS proposé par (Herzig *et al.*, 2020), utilise d'une manière originale l'architecture BERT (Devlin *et al.*, 2019), pour proposer un nouveau traitement de cette tâche ; il donne actuellement les meilleurs résultats.
Cependant toutes ces ressources ne sont conçues que pour un usage qui concerne la langue anglaise. Par suite, l'on peut se demander dans quelle mesure une nouvelle collecte de données est nécessaire pour obtenir un modèle équivalent à TAPAS pour le français. C'est pourquoi nous proposons ici une traduction du jeu de données proposé par (Zhong *et al.*, 2017), ainsi que son évaluation sur une version ré-entraînée du modèle TAPAS, afin de déterminer si ce jeu de données obtenu par traduction d'une ressource anglophone est suffisant pour obtenir un modèle de traitement de questions en langage naturel sur des données tabulaires en français. Plus précisément, nous souhaitons établir dans quelle mesure ce modèle ré-appris sur les données traduites produit des résultats comparables à ceux obtenus pour l'anglais sur une tâche équivalente.

2 Présentation des jeux de données

Afin de réaliser la tâche de traduction du langage naturel en une expression SQL, plusieurs jeux de données peuvent être exploités :

- **WIKITableQuestion** (Pasupat & Liang, 2015) est un jeu de données composé de questions sur des tables HTML issues de Wikipedia auxquelles sont associées des questions complexes réalisées par des humains à qui il a été demandé de créer, suivant une table donnée, des questions complexes dont la réponse nécessite plusieurs opérations sur la table (agrégation, comparaisons, superlatifs, opérations mathématiques). Au total il comprend 22 033 questions sur 2 108 tables.
- **SQA** (Iyyer *et al.*, 2017) : cet ensemble de données a été construit en demandant à des humains de décomposer un sous-ensemble de questions hautement compositionnelles de WIKITQ, où chaque question décomposée résultante peut être renseignée par une ou plusieurs cellules d'une table SQL. L'ensemble final se compose de 6 066 séquences de questions avec 2,9 questions par séquence en moyenne.
- **WikiSQL** (Zhong *et al.*, 2017) : ce jeu de données se concentre sur la traduction de texte en SQL. Il a été construit en demandant à des humains de paraphraser une question basée sur un modèle en langage naturel, deux autres étant invités à vérifier la qualité des paraphrases proposées. Le résultat est un ensemble de 80 654 questions sur 24 241 tables issues de Wikipédia.

Entre ces trois jeux de données, notre choix s'est porté sur WikiSQL, car c'est le plus important d'un point de vue quantitatif, mais aussi parce qu'il fait office de benchmark pour cette tâche, étant souvent cité en ce sens ((Baik *et al.*, 2019), (Lyu *et al.*, 2020)). De plus dans leur article présentant le modèle, (Herzig *et al.*, 2020) ont pu tester l'apprentissage par transfert de WIKISQL vers un autre jeu de données avec un certain succès. Notre expérience étant fondée sur cette application de l'apprentissage par transfert sur des données traduites, le choix de ce jeu de donné semble justifié.

De la même manière que (Kabbadj, 2018) ont proposé une traduction du jeu de données SQuAD en

utilisant l'API de google traduction, nous proposons une version traduite du jeu de données WikiSQL en utilisant cette même API. Cette tâche de traduction comporte trois étapes :

— la traduction de la question en langage naturel de l'anglais vers le français
— la traduction des entêtes des colonnes de la table lorsque cela est nécessaire
— le remplacement des entêtes dans les requêtes SQL.

Le résultat de cette étape de traduction est illustré par les exemples présentés dans les tableaux 1 et 2.

	Original
Question	What is the UNGEGN, when the Value is 10 000 ?
Headers	['Value', 'Khmer', 'Word Form', 'UNGEGN', 'ALA-LC', 'Notes']
SQL	'SELECT UNGEGN FROM table WHERE Value = 10 000'

TABLE 1 – Exemple d'une question du jeu de données WikiSQL, ainsi que les entêtes de la table associée, et la requête SQL correspondante

	Traduction
Question	Qu'est-ce que l'UNGEGN, lorsque la valeur est de 10 000 ?
Entêtes	['Valeur', 'Khmer', 'Forme lexicale', 'UNGEGN', 'ALA-LC', 'Notes']
SQL	'SELECT UNGEGN FROM table WHERE Valeur = 10 000'

TABLE 2 – Exemple du résultat d'une traduction d'un item du jeu de données WikiSQL

Par ailleurs, nous avons respecté la partition du jeu de données original, à savoir, 56355 (70%) questions dans le jeu d'apprentissage, 8421 (10%) dans le jeu de validation et 1578 (20%) dans le jeu de test.

De plus, on peut appréhender la complexité des requêtes du jeu de données de deux manières : d'une part en observant la longueur des requêtes en langage naturel comme présenté dans le diagramme en figure 1, et d'autre part en observant les proportions des différents agrégateurs dans le jeu de données comme dans le diagramme en figure 2.

On remarque alors que les requêtes en langage naturel sont assez courtes (autour d'une dizaine de mots). Cette distribution est semblable à celle du jeu de données anglophone présenté par (Zhong *et al.*, 2017). De plus, on peut voir dans le diagramme en figure 2, que la majorité des requêtes SQL n'utilisent pas de fonctions d'agrégat, et ne s'assimilent donc qu'à une simple sélection sur la table. Ainsi ce jeu de données se caractérise par des requêtes d'une relative simplicité.

2.1 La traduction des données

Notre choix s'est porté sur les systèmes de Google Translate afin de suivre le même protocole de traduction que celui proposé par (Kabbadj, 2018) pour passer des données de Squad à SquadFr. De plus une étude récente réalisée par (Aiken, 2019) montre que l'outil est suffisamment performant pour notre cas d'usage. Toutefois, il serait intéressant d'étudier plus spécifiquement l'impact de la traduction sur les performances du modèle et ceci pourra être envisagé dans des travaux futurs.

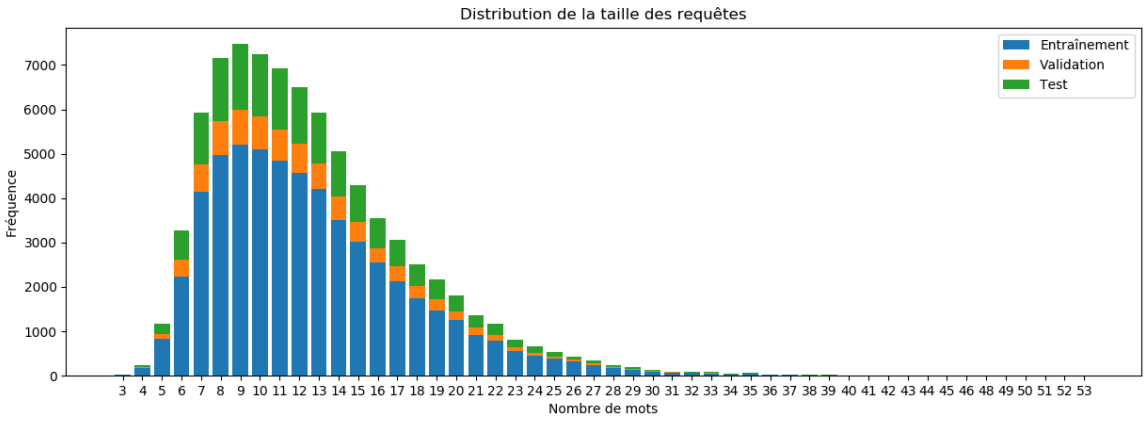


FIGURE 1 – Diagramme représentant la distribution des requêtes en français selon leur longueur

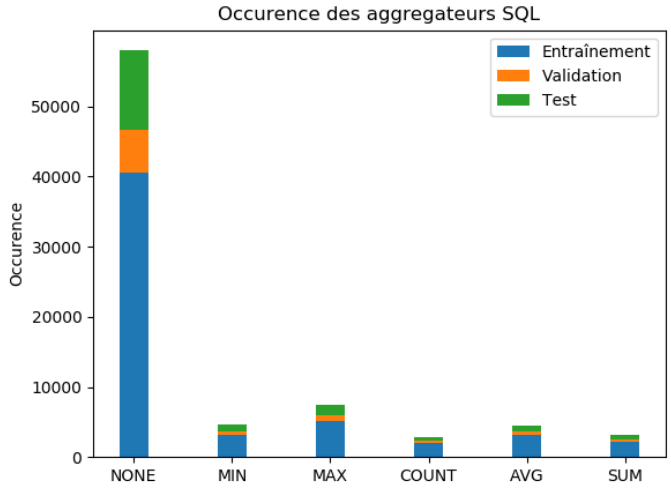


FIGURE 2 – Diagramme représentant la proportion de chaque agrégateur SQL dans les différentes partitions du jeu de données

3 Présentation du modèle TAPAS

Avec les progrès récents de l'apprentissage profond appliqué à la compréhension du langage naturel, des prototypes de logiciels grand public cherchant à intégrer une interrogation en langage naturel pour assurer une interaction homme-machine plus naturelle ont vu le jour. Les services proposés restent cependant extrêmement limités car ils ne prennent en compte que des structures de données spécifiques, et les interactions restent encore très éloignées d'une interaction en langage naturel. Dans le même temps, de nouvelles architectures neuronales permettent de franchir des étapes importantes pour déterminer les réponses à donner à des questions exprimées en langue naturelle. En particulier, les architectures à base de transformers telles que le BERT de (Devlin *et al.*, 2019) ont apporté des progrès notables. Pourrions-nous tirer parti de ces avancées pour interagir avec les données tabulaires ? Récemment, Google Research (Herzig *et al.*, 2020) a dévoilé TAPAS (Table parser), un modèle basé sur l'architecture BERT qui traite les questions et réponses pour des ensembles de données tabulaires.

Au lieu de créer un modèle contraint à une structure de table spécifique, Google a fait le choix d'une approche plus globale en créant un réseau de neurones adapté à toute forme de jeu de données tabulaires. Son modèle TAPAS réutilise l'architecture de l'encodeur BERT, en y ajoutant des plongements supplémentaires. L'ajout le plus notable au modèle de base BERT est l'intégration d'informations supplé-mentaires pour l'encodage de l'entrée textuelle. Tapas exploite les incorporations apprises pour les index de ligne et de colonne ainsi que pour un index de rang spécial qui représente l'ordre des éléments dans les colonnes numériques. L'architecture obtenue surpasse actuellement les autres modèles pour l'interrogation en langage naturel de données tabulaires.

Le modèle TAPAS est assez similaire à BERT mais il en diffère par l'ajout à son *tokenizer* des plongements des positions relatives, ainsi que sept tokens modélisant les tables. TAPAS est pré-appris sur une tâche de modèle masqué [1], à l'aide de millions de tables venant de la version anglaise de Wikipedia et les textes correspondants.

Enfin, TAPAS est entraîné plus finement sur une tâche des réponses aux questions. Le modèle cherche alors à prédire deux choses : les cellules correctes associées à la réponse et l'agrégateur correspondant.

4 Application et résultats

4.1 Apprentissage du modèle

Outre les ajouts et modifications apportés au modèle BERT originel décrit précédemment, TAPAS suit un protocole d'apprentissage en trois étapes qui s'appuie sur plusieurs jeux de données. La première étape consiste en un pré-entraînement sur un jeu de données de 6,2 millions de tables anglophones extraites de Wikipedia suivant le modèle de masquage proposé dans BERT (Devlin *et al.*, 2019). Le but ici est d'initialiser l'entraînement du modèle à partir du contexte constitué des éléments qui composent la table traitée : la tâche consiste à retrouver certains éléments masqués de ce contexte.

Cette étape de pré-apprentissage est suivie d'une étape d'ajustement fin (fine tuning) qui finalise

1. Masked language modeling (MLM)

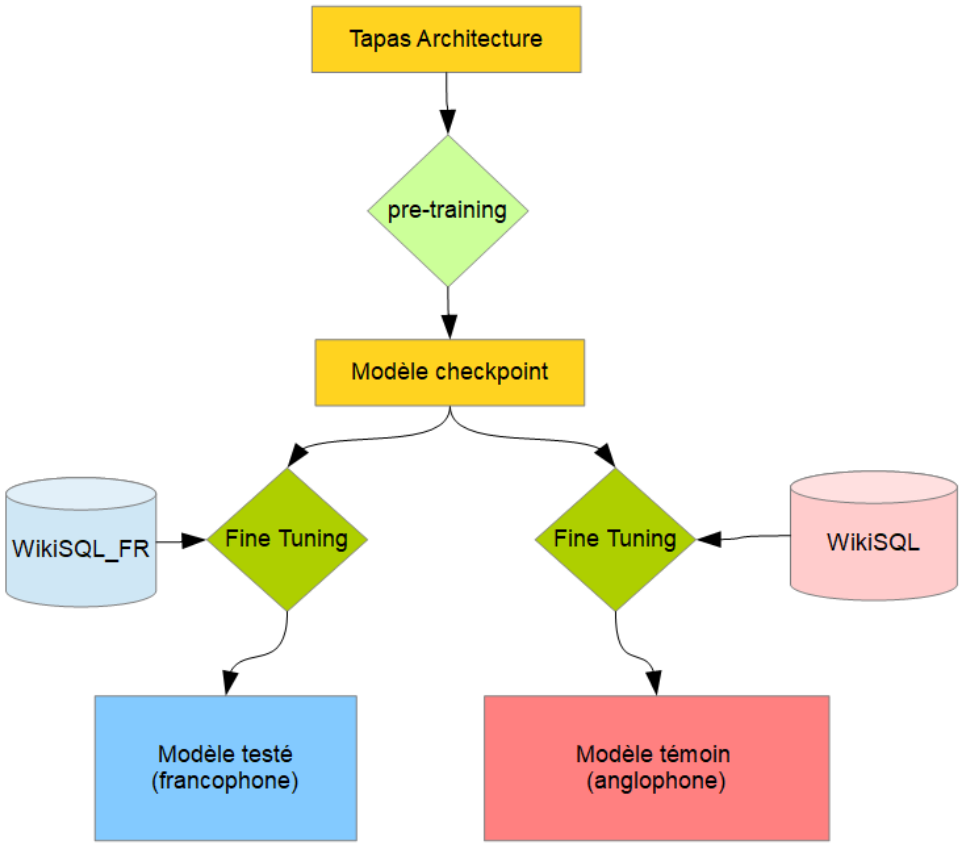


FIGURE 3 – Schéma représentant le protocole d'obtention des modèles

l'apprentissage du modèle sur une tâche spécifique. Les données utilisées pour cet entraînement sont alors similaires au jeu de tests pour évaluer le modèle final obtenu.

Dans leurs travaux les plus récents sur le modèle (Eisenschlos *et al.*, 2020), les auteurs ont ajouté une étape intermédiaire d'apprentissage. Pour notre expérimentation, nous sommes partis de ce modèle ayant bénéficié de ce pré-entraînement supplémentaire.

4.2 Expérimentation sur les données francophones

Le but de l'expérimentation est de déterminer dans quelle mesure ce nouveau jeu de données francophones impacte les performances du modèle TAPAS. Ainsi l'un des modèles TAPAS entraîné sur WikiSQL pour la langue anglaise sert de modèle témoin. À cela on compare un modèle suivant la même architecture et ayant reçu le même pré-entraînement, mais dont l'entraînement fin a été réalisé sur le nouveau jeu de données francophones. Les deux jeux de données sont alors testés sur leurs jeux de données respectifs. Un schéma récapitule ce processus en figure 3, les indicateurs de sont les mêmes que ceux décrits dans l'article originel de Tapas.

Ainsi afin d'avoir les conditions expérimentales les plus similaires entre ces deux modèles, nous avons pris comme point de contrôle (checkpoint) d'apprentissage le modèle *base* de TAPAS que nous avons entraîné sur les jeux de données respectifs sur 100000 et 200000 pas de batch 4 pour un total de 400000 et 800000 mises à jour. On obtient alors les résultats en tableau 3. Les méthodes de test

sont les mêmes que celles utilisées par TAPAS, et les modèles sont testés sur les mêmes données que celles utilisées pour l'apprentissage fin.

	Modèle anglophone	Modèle francophone
100000 pas dev/test ex(acc)	0.8248/0.7993	0.5366 / 0.5195
200000 pas dev/test ex(acc)	0.8246 / 0.7979	0.6028 / 0.5864
résultats finaux [2]dev/test ex(acc)	0.8859	0.5967/0.5774

TABLE 3 – Exactitude (accuracy) sur le jeu de tests et développement des deux modèles anglophone et francophone.

On peut alors voir plusieurs différences entre les résultats des deux modèles. Tout d'abord le modèle francophone semble avoir, après 200000 pas, des résultats 20% inférieurs à ceux du modèle anglophone. De plus le modèle de langue française semble bien plus sensible à cette dernière phase d'apprentissage que le modèle anglophone. Cette différence dans la vitesse de convergence dans la deuxième phase peut s'expliquer par le changement de langue entre la phase de pré-apprentissage et la phase d'apprentissage fin, les résultats finaux présentés pour notre modèle francophone ayant atteint une valeur asymptotique.

5 Conclusion et travaux futurs

Si les résultats propres de l'expérience ne permettent pas d'en déduire pour l'instant une réelle efficacité du modèle en situation réelle, ils permettent néanmoins d'encourager cette approche de traduction de jeu de données. En effet on peut espérer de meilleurs résultats avec un apprentissage plus long, ou un jeu de données traduit et relu, ce qui allégerait la tâche d'expertise humaine nécessaire à l'obtention d'un jeu de données pour cette tâche de traitement de requêtes en français sur des données tabulaires.

De plus ces résultats nous éclairent sur l'usage de l'apprentissage fin, qui permet d'une part d'adapter plus facilement de larges modèles à de nouvelles langues, et d'autre part d'éviter de travailler sur des modèles complets souvent très volumineux ; dans notre cas les données de pré-apprentissage forment un corpus de 6.2 millions de tables.

Enfin, des résultats complémentaires permettront d'affiner les conclusions et perspectives de cette étude, comme sur l'influence de la traduction sur les performances d'exactitude. On peut aussi imaginer un modèle suivant la même architecture mais prenant en compte des données francophones sur chaque étape de l'apprentissage.

Références

AIKEN M. (2019). An updated evaluation of google translate accuracy. *Studies in Linguistics and Literature*, **3**, p253. DOI : 10.22158/sll.v3n3p253.

2. Tels que présentés dans l'article de (Eisenschlos *et al.*, 2020), après entraînement sur TPU.

BAIK C., JAGADISH H. V. & LI Y. (2019). Bridging the semantic gap with sql query logs in natural language interfaces to databases. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, p. 374–385.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding.

EISENSCHLOS J. M., KRICHENE S. & MÜLLER T. (2020). Understanding tables with intermediate pre-training.

HERZIG J., NOWAK P. K., MÜLLER T., PICCINNO F. & EISENSCHLOS J. (2020). TaPas : Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4320–4333, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.398.

IYYER M., TAU YIH W. & CHANG M.-W. (2017). Search-based neural structured learning for sequential question answering. In *ACL (1)*, p. 1821–1831.

KABBADJ A. (2018). Something new in french text mining and information extraction (universal chatbot) : Largest qa french training dataset (110 000+). [Online ; posted 11-November-2018].

LYU Q., CHAKRABARTI K., HATHI S., KUNDU S., ZHANG J. & CHEN Z. (2020). Hybrid ranking network for text-to-sql.

PASUPAT P. & LIANG P. (2015). Compositional semantic parsing on semi-structured tables. 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. 1470–1480, Beijing, China : Association for Computational Linguistics. DOI : 10.3115/v1/P15-1142.

RAJPURKAR P., JIA R. & LIANG P. (2018). Know what you don't know : Unanswerable questions for squad.

WANG A., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. R. (2019). Glue : A multi-task benchmark and analysis platform for natural language understanding.

ZHONG V., XIONG C. & SOCHER R. (2017). Seq2sql : Generating structured queries from natural language using reinforcement learning. *CoRR*, **abs/1709.00103**.

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

Enjeux liés à la détection de l'ironie

Samuel Laperle[1]
(1) Université du Québec à Montréal, Montréal, Canada
laperle.samuel@courrier.uqam.ca

RÉSUMÉ

L'ironie verbale est un type de discours difficile à détecter automatiquement. En créant des ponts entre les recherches en linguistique et en informatique sur cette question, il est possible de souligner des caractéristiques importantes permettant de faciliter ce type de tâche. Dans cet article, il sera question du rapport entre la définition de ce phénomène et son adéquation avec l'élaboration de corpus d'entraînement..

ABSTRACT

Challenges of automatic irony detection

Verbal irony is a type of speech that is difficult to detect. By building bridges between linguistics and computer science research on this question, it is possible to highlight important characteristics that facilitate this type of task. This paper will focus on the relationship between the definition of this type of discourse and its adequacy with the development of training corpus.).

MOTS-CLÉS : Ironie, détection automatique, linguistique.

KEYWORDS: Irony, automatic detection, linguistics.

1 Introduction

La question de la détection de l'ironie verbale par des systèmes informatiques est un enjeu économique et marketing ayant généralement comme objectif d'améliorer les algorithmes d'analyse de sentiment qui permettent de traiter d'énormes quantités de données issues des médias sociaux (Eke *et al.*, 2020; Strapparava *et al.*, 2011). L'ironie est un type de discours compliqué à définir. De ce fait, les algorithmes proposés pour tenter de détecter adéquatement ce type de discours doivent jongler avec des variables difficilement implémentables. Par conséquent, on retrouve une rupture entre le travail linguistique sur la caractérisation de l'ironie verbale et ses traitements computationnels. Ce travail tentera de mettre en relief l'écart présent dans ces deux cadres de recherche concernant cet enjeu.

Concrètement, la question centrale de ce travail est de déterminer au niveau théorique les limitations des méthodes computationnelles de détection de l'ironie verbale. D'abord, il sera question de faire une présentation des différentes définitions de l'ironie verbale proposée en linguistique. Ensuite, certains types de méthodes de détection automatique proposées en TAL seront présentés. Avec ces deux perspectives, il sera possible de les confronter pour déterminer, sur la base des théories linguistiques, les différents enjeux propres à la définition de l'ironie. Par exemple, il sera question de la confusion entre les concepts de sarcasme et d'ironie, de la construction des corpus d'entraînement et de la polarité des énoncés ironiques.

2 Définir l'ironie

Pour Kerbrat-Orecchioni (1978), l'ironie serait un acte illocutoire permettant de se moquer d'une cible. Elle qualifie plus particulièrement l'ironie verbale comme étant « la mise en relation entre deux niveaux sémantiques littéral et figuratif attachés à une même séquence signifiante ». De façon similaire, Grice (1975) caractérise l'ironie comme étant une transgression de la maxime de qualité qui stipule qu'un locuteur ne devrait pas dire ce qu'il croit être faux.

Bien que ces définitions soulèvent des points primordiaux, Wilson & Sperber (1992) soulignent certaines de leurs limitations. Concrètement, elles négligent trois types d'ironie et surgénéralisent sur des types de discours non ironiques, soit respectivement les litotes ironiques, les citations ironiques, les interjections ironiques et les mensonges éhontés. De ce fait, Wilson & Sperber (1992) quant à eux, définissent l'ironie par la théorie de l'écho, qui s'appuie sur la distinction entre mention et usage. Pour eux, l'ironie serait un type de citation indirecte transmettant l'attitude d'un locuteur concernant une cible. Parallèlement, (Clark & Gerrig, 1984) définissent l'ironie par la théorie du faire-semblant. Cette proposition met l'accent sur la relation entre le locuteur et l'interlocuteur plutôt que sur les processus interprétatifs de ce dernier. Pour eux, le locuteur joue un rôle et l'interlocuteur doit être en mesure de déceler la mascarade.

Les propositions de Grice (1975), Wilson & Sperber (1992) et Clark & Gerrig (1984) s'imposent comme des incontournables. Elles négligent néanmoins certains points importants. On n'y retrouve aucune référence à l'aspect évaluatif caractéristique de l'ironie.

Pour Alba-Juez & Attardo (2014), le spectre d'attitudes pouvant être transmises par l'ironie verbale serait large. Par exemple, on peut retrouver des énoncés ironiques qui transmettent une attitude négative comme en (1), une attitude positive comme en (2) ou, même, une attitude neutre comme en (3).

1. Quelle belle partie ! (exclamé suite à une défaite)
2. Quelle triste partie ! (exclamé suite à une victoire)
3. C'était une partie. (exclamé suite à une partie s'étant rapidement terminée)

De plus, un énoncé peut transmettre différentes attitudes à différentes cibles à la fois. Prouvant ce point, Alba-Juez & Attardo (2014) proposent l'exemple (4) où une actrice doutant de son talent dirait à son ami :

4. A : Je suis un échec. Je ne réussirai jamais à percer dans le monde du théâtre. Je suis une actrice médiocre.

Pour qu'après la réception d'un prix soulignant son talent, cet ami lui réponde (5) :

5. Félicitations, mon amie ! Tu es une actrice médiocre. Je ne sais pas comment ils ont pu te donner ce prix !

D'un côté, on note que le locuteur en (5) transmet une évaluation positive du talent d'actrice de son amie tout en émettant une appréciation négative du jugement négatif qu'elle s'auto-imposait. Ces variations dépendent des cibles visées par le locuteur et du contexte de savoir partagé entre les individus.

Ce contexte joue aussi un rôle primordial dans la production et l'interprétation d'un énoncé ironique. Cette capacité ne dépend pas seulement de la relation entre un mot ou une expression avec l'ensemble d'une situation ou d'un texte (Martini *et al.*, 2018). Elle dépend aussi de notre faculté à nous mettre à la place d'autrui (Nilsen *et al.*, 2011). Les objectifs communicationnels varient et s'adaptent aux informations que nous collectons à travers nos conversations. Ainsi, comme le rapporte Gibbs (2000), nous sommes plus prompts à utiliser l'ironie dans des contextes sociaux où nous connaissons bien nos interlocuteurs, car ces derniers arriveraient plus facilement à déterminer adéquatement les attitudes véritables que nous tentons de communiquer.

Ces différents travaux soulignent des caractéristiques importantes de l'ironie verbale que devront prendre en compte les algorithmes de détection automatique de ce type de discours.

3 Détection l'ironie

3.1 Méthodes à base de règles

Les informations langagières présentes sur Internet étant textuelles, la plupart des algorithmes fonctionnent en se basant principalement sur la présence d'ironie verbale exprimée sous cette modalité. Par exemple, Kreuz & Caucci (2007) proposent d'évaluer les expressions considérées comme étant stéréotypiques de l'ironie verbale pour les ajouter à des algorithmes de détection. Pour ce faire, des participants devaient lire des énoncés en anglais préalablement collectés par les chercheurs et évaluer leur niveau d'ironie. En collectant ainsi environ 100 énoncés, ils notent certains traits lexicaux plus fréquents lors de l'expression de ce type de discours comme des interjections, des expressions convenues (thanks alot, good job), des questions rhétoriques et de la répétition. Allant dans le même sens, Bouazizi & Ohtsuki (2015) proposent d'utiliser comme marqueurs lexicaux de l'ironie la présence de mots peu communs en termes de fréquence ou la présence d'énoncés, eux aussi, considérées comme prototypiques de ce type de discours (P.ex. : « love [pronoun] when » ou « [pronoun] be [adverb] funny »). Ils justifient ce choix en soulignant que l'ironie peut être utilisée comme une façon d'éviter de donner une réponse claire à une question. Se faisant, le locuteur emploie des phrases plus longues, plus complexes et, ainsi, utilise des expressions moins fréquentes. Ce raisonnement fait écho à une des raisons possibles derrière l'utilisation de l'ironie proposée par Jorgensen (1996). Pour ce dernier, ce type de discours pourrait être utilisé pour transmettre une critique sans qu'elle soit perçue comme étant trop directe ou négative.

Comme Attardo (2000) le mentionne, l'ironie est un type de discours pragmatique. De ce fait, se baser sur des informations exclusivement lexicales peut s'avérer problématique. Conséquemment, Joshi *et al.* (2015) ont mis au point un algorithme de détection basé sur la présence d'incongruité textuelle explicite ou implicite. Dans le premier cas, on peut soupçonner qu'un énoncé soit sarcastique en se basant sur un lexique (Lingpipe SA system (Alias-I, 2014)) s'il comporte deux mots de valences différentes comme dans l'énoncé « j'aime être malade », où « aimer » porte une valence positive et

« malade » en porte une qui est négative. Cette catégorisation ne peut pas s'appliquer dans des cas d'incongruité implicite comme dans une expression du genre « J'aime tellement ce repas que je l'ai donné à mon chien ». Dans cette dernière, les éléments lexicaux « donner à son chien » ne sont pas explicitement négatifs et ne créent pas d'incongruité avec l'expression positive « j'aime ». Pour rendre compte de ce type d'expressions convenues, les chercheurs proposent de créer un système à base de règles impliquant divers types de phrases typiques portant une polarité implicite. En plus de ces

deux paramètres, l'architecture des chercheurs tient en compte des traits lexicaux soit des unigrames, la présence de lettres majuscules, d'émoticônes ou de rires et de ponctuations particulières comme des points d'exclamation excessifs. Pour tester ce système, ils ont utilisé trois bases de données. La première, Tweet-A, contient 5208 tweets non sarcastiques et 4170 tweets sarcastiques collectés grâce à la présence de #sarcasm. Le deuxième, Tweet-B, contient 2278 tweets non sarcastiques et 506 tweets sarcastiques. Ces derniers sont tirés du travail préalable de Riloff *et al.* (2013). Le troisième jeu de données, Discussion-A, provient d'un corpus préalablement créé par Walker *et al.* (2012). Au total, il contient 1502 messages littéraux et 752 messages sarcastiques. Parmi ceux-ci, Joshi *et al.* (2015) en gardent 752 littéraux et 752 sarcastiques. Ces messages sont issus de forums de discussion en ligne.

3.2 Méthodes à base d'apprentissage machine

en termes d'apprentissage machine, Poria *et al.* (2016) proposent un modèle basé sur un réseau neuronal convolutif (RNC) pré entraîné pour extraire des traits concernant les sentiments, les émotions et la personnalité du locuteur. Concrètement, selon Poria *et al.* (2016), grâce à ce type de réseau, il est possible de former un vecteur englobant l'ensemble de traits locaux d'un énoncé lui permettant de créer une représentation adéquate du contexte lexical. Ils ont testé leur algorithme sur trois jeux de données. Le premier, créé par Ptáček *et al.* (2014), contient un nombre balancé de tweets sarcastiques (50 000) et non sarcastiques (50 000) en anglais. Le deuxième, aussi créé par Ptáček *et al.* (2014), contient un nombre déséquilibré de tweets sarcastiques (25 000) et de tweets non sarcastiques(75 000) en anglais. Le troisième, issu du site internet The Sarcasm Detector, [1] contient 20 000 tweets sarcastiques et 100 000 tweets non sarcastiques en anglais. Parmi ceux-ci, Poria et al. (2016) en ont collecté de façon aléatoire 10 000 sarcastiques et 20 000 non sarcastiques. Sur l'ensemble de ces jeux de données, ils arrivent à des F1 supérieurs à 0.9 de détection de l'ironie verbale.

De leur côté, Ghosh & Veale (2016) utilisent une conjonction de différents types de réseaux neuronaux, soit un réseau de neurones composé d'un RNC, suivi d'un réseau de neurones récurrent (RNR), d'un long-short term memory (LSTM) et d'un réseau de neurones profond (RNP). La première couche de cette architecture est celle des données langagières contenues dans un tweet qui est vectorisé. Ensuite, le résultat de ce traitement passe par une couche du RNC qui permettrait d'extraire des séquences de mots importants pour la détection en éliminant les variations de fréquences. Ce processus fournit à la couche du LSTM les données adéquates. Ce dernier serait en mesure de créer une représentation sémantique grâce à la présence d'un module temporel permettant d'entreposer des informations contextuelles. Finalement, les données sont traitées par le RNP. Pour tester leur modèle, ils ont utilisé deux jeux de données préalablement créés par Tsur *et al.* (2010) et Riloff *et al.* (2013). Le premier jeu de données comprend 471 critiques en anglais de produit en vente sur le site Amazon classées comme étant sarcastique et de 5020 critiques non sarcastiques. Le second jeu de données est composé de 693 tweets sarcastiques et 2 307 tweets non sarcastiques en anglais L'ensemble de ces couches arrive à un score de précision de 0.919, un score de rappel de 0.923 et un score-f de 0.921. Malgré ces bons résultats, Ghosh & Veale (2016) notent tout de même que leur système de détection n'est pas en mesure de classifier une expression ironique comme

« Thank God it's Monday ! » bien qu'il soit en mesure de détecter l'ironie dans une expression « I just love Mondays ! ».

1. thesarcasmdetector.com

4 Limitations

4.1 Problèmes de définition : ironie ou sarcasme

Un des enjeux centraux relatifs à la détection automatique de l'ironie se situe au niveau de la définition même de ce concept. La majorité des travaux concernant ce sujet semble éviter d'aborder cette question. Paradoxalement, la façon dont on choisit d'appréhender ce problème influence nécessairement les méthodes employées pour arriver à une détection adéquate de ce type de discours. Lorsqu'on en trouve, elles sont essentiellement superficielles. Par exemple, Carvalho *et al.* (2009) caractérisent l'ironie verbale ainsi : *as the rhetorical process of intentionally using words or expressions for uttering a meaning different (usually the opposite) from the one they have when used literally*. Dans le même sens, Van Hee (2017) propose de décrire l'ironie comme "*an evaluative expression whose polarity (i.e. positive, negative) is inverted between the literal and the intended evaluation, resulting in an incongruence between the literal evaluation and its context*". Ces définitions ressemblent à celles proposées par Kerbrat-Orecchioni (1978) et Grice (1975). Par conséquent, percevoir l'ironie verbale comme étant une négation du sens littéral ou comme un rapport de polarité inverse néglige un ensemble important d'énoncés ironiques s'exprimant autrement.

Par ailleurs, dans la littérature sur la détection automatique de l'ironie, on retrouve fréquemment une juxtaposition de ce terme avec celui désignant le sarcasme. On considère parfois ces deux concepts comme étant interchangeables. Par exemple, Davidov *et al.* (2010) écrivent : *sarcasm (also known as verbal irony) is a sophisticated form of speech act in which the speakers convey their message in an implicit way*". D'autres fois, on présente l'ironie comme étant une supracatégorie pouvant contenir des énoncés sarcastiques (sans expliquer comment) comme dans cet exemple (Farías *et al.*, 2016) : "*irony is here considered an umbrella term that also covers sarcasm*".

Au niveau de cette distinction difficile à faire, Van Hee (2017) rapporte que "*researchers do not differentiate between irony and sarcasm [because they observed] a shift in meaning between the two terms. Over time, the term 'sarcasm' seems to have gradually replaced what was previously designed by 'irony' (Nunberg, 2001)*".

Les points apportés par (Van Hee, 2017) sont valides. Toutefois, ils font fi de tout un pan de la littérature qualifiant le sarcasme, qui possède une connotation plus négative que le terme « ironie » (Kreuz & Caucci, 2007). De plus, on n'évalue pas comment ces problèmes affectent indéniablement les processus de détection. Il existe un rapport évident entre la définition du phénomène que l'on souhaite détecter et la façon prévue pour y arriver. De ce fait, une attention particulière devrait être accordée à cette question.

Goddard (2018) souligne ce type de problème circulaire au niveau de la définition terminologique d'un concept :

> (i) One starts with ordinary English words, poorly defined or undefined, then (ii) "technicalizes" them and extends their range, often making some formal adjustments along the way (iii) Subsequently [..], different scholars begin to employ the terms, often using them in slightly different ways from the original authors. (iv) Scholarly debate begins about what the new terms mean or should mean.

S'il est indéniable qu'« ironie » et « sarcasme » sont liés, il est faux de dire que ces deux termes désignent la même chose. En effet, par exemple, en termes d'usage, on pourra dire d'une situation

qu'elle est ironique, mais jamais sarcastique. Allant dans ce sens, Sulis *et al.* (2016) ont évalué un corpus contenant 10 000 tweets pour évaluer les différences caractéristiques entre les tweets contenant l'expression #irony, #sarcasm et #not. Parmi ces divergences notables, ils rapportent qu'au niveau affectif, les tweets contenant l'expression #irony contiennent moins de mots associés à la joie et l'anticipation que les tweets contenant l'expression #sarcasm. Au contraire, le #irony serait plus associé à des sentiments comme la colère, la tristesse et la peur. En utilisant des lexiques (Affective Norms for English Words, Dictionary of Affective Language) recensant des valeurs comme le niveau d'imagerie, d'activation générale et émotive et de dominance, Sulis *et al.* (2016) soulignent aussi que les énoncés comportant le mot clé #irony seraient plus subtils que ceux contenant #sarcasm. Se faisant, il est inadéquat d'utiliser les termes sarcasme et ironie de façon interchangeable. Si ces concepts sont difficiles à distinguer, on remarque néanmoins des différences concrètes au niveau de leurs usages.

4.2 La polarité d'un énoncé

Comme mentionné plus haut, la polarité d'un énoncé ironique n'est pas systématiquement négative. Alba-Juez & Attardo (2014) ont démontré les différents cas de figure où un énoncé ironique peut avoir une polarité positive, négative et neutre. De plus, parfois, un même énoncé peut être positif envers une cible tout en étant négatif envers une autre. Ainsi, des algorithmes comme celui de Joshi *et al.* (2015) présenté plus haut négligeront nécessairement des énoncés ironiques comme celui présenté en (5) dans la section 2.

Les énoncés ironiques neutres sont particulièrement problématiques de par leur nature. Alba-Juez & Attardo (2014) les définissent comme étant des énoncés qui ne visent, ne critiquent et qui ne vantent personne ni rien de particulier. Bien que ce type d'énoncé comporte une certaine valeur évaluative, cette dernière est loin d'être positive ou négative. La motivation derrière ce type d'énoncé serait avant tout de faire preuve d'humour. Dans l'exemple (6), il est difficile de catégoriser adéquatement l'attitude voulant être transmise.

6. Ce courriel est plus long qu'à l'habitude parce que je n'avais pas le temps d'en écrire un plus court.

Ainsi, pour Alba-Juez & Attardo (2014), l'ironie ne dépend pas de son caractère évaluatif, mais plutôt de l'inférence contradictoire qui en découle. Se faisant, en (6), c'est de la contradiction entre la longueur du courriel et l'excuse en général qui permet l'émergence d'une interprétation ironique de l'énoncé.

4.3 Corpus

Généralement, les corpus d'énoncés ironiques se construisent de deux façons (Farías *et al.*, 2016). D'un côté, on retrouve les énoncés ironiques explicitement indiqués comme tels par les locuteurs les produisant. Concrètement, sur Twitter, on aurait des tweets contenant des hashtags comme #sarcasm, #sarcastic ou #nottrue. Derrière l'élaboration de ce type de corpus, on prend pour acquis que le locuteur serait le mieux placé pour savoir si un énoncé qu'il produit est ironique ou non. De l'autre côté, on trouve des corpus basés sur des accords interjuges. La qualité de ces derniers dépend évidemment de facteurs internes aux juges ayant catégorisé les éléments constitutifs du corpus.

Idéalement, les corpus construits grâce à la présence de hashtag passent par la suite par une équipe d'annotateurs jugeant si les tweets sont réellement perçus comme étant sarcastiques. Ce deuxième tri dépend fortement de la définition de sarcasme ou d'ironie utilisée par l'équipe de chercheurs.

Au-delà de ces aspects, un autre problème dans les études mentionnées plus haut est l'inadéquation entre les définitions de l'ironie que certains chercheurs proposent et les corpus d'énoncés ironiques utilisés.

Par exemple, Kreuz & Caucci (2007), en tentant de vérifier s'il existe des termes lexicaux permettant de caractériser l'ironie, ne proposent pas de définir ce terme. Néanmoins, ils construisent tout de même un corpus d'analyse en sélectionnant sur Google Books des énoncés suivis de l'expression "said sarcastically" en prenant soin de retirer cette expression avant de les présenter à leurs participants. Donc, on ne sait pas sur quels critères se basent ces derniers pour émettre leurs jugements. De plus, comme le mentionnent Kreuz & Caucci (2007), il est possible qu'un auteur utilise l'expression "said sarcastically" comme synonyme de "said jokingly" ou "said angrily", entraînant nécessairement des biais importants concernant le type d'énoncé qu'on retrouve dans un tel corpus.

Dans le même ordre d'idée, dans le travail de Bouazizi & Ohtsuki (2015), on définit le sarcasme comme étant "*a special form of irony by which the person conveys implicit information, usually the opposite of what is said, within the message he transmits*". Cependant, on ne sait pas comment cette conceptualisation se traduit au niveau de l'élaboration des corpus utilisés. Dans ce cas-ci, ces derniers sont composés de tweets comportant l'expression "#sarcasm" pour, par la suite, être revérifiés par les chercheurs. Outre leur définition du sarcasme mentionné plus haut, il nous est donc impossible de savoir sur quelles bases sont fondés ces critères de sélection.

Cette relation entre la définition du concept étudié, l'élaboration d'un algorithme de détection automatique et le choix des éléments constituant les jeux de données permettant d'extraire des traits ou de tester leur justesse est particulière. Si l'élaboration de l'algorithme de détection de l'ironie dépend de la caractérisation de ce type de discours par les chercheurs, l'élaboration du corpus test doit nécessairement refléter cet aspect. Dans le cas contraire, on peut s'attendre à la présence de faux positifs ou de faux négatifs.

4.4 Problèmes concernant les algorithmes d'apprentissage machine

Les approches se basant sur les techniques d'apprentissage machine nous offrent des résultats variables concernant la détection automatique de l'ironie. Comme pour les méthodes à base de règles, elles réussissent généralement bien dans les conditions tests. Néanmoins, de par la nature de ces méthodes, il nous est impossible de savoir concrètement ce que ces algorithmes "apprennent". Aussi, on sait peu de choses sur la possibilité de généraliser ces apprentissages à un milieu plus écologique. À ce sujet, Wallace (2015) dit :

> Current machine learning methods rely too heavily on shallow, unstructured, syntactic modelling of text to consistently discern ironic intent. Irony detection is an interesting machine learning problem because, in contrast to most text classification tasks, it requires a semantics that cannot be inferred directly from word counts over documents alone.

En effet, récemment, dans une tâche partagée axée sur la détection de sarcasme, Ghosh *et al.* (2020) évaluent différents modèles proposés. Dans cette tâche, les participants devaient proposer un algorithme qui serait en mesure de déterminer, en ayant accès au contexte conversationnel nécessaire,

la présence de sarcasme dans un énoncé. Les meilleurs résultats ont été obtenus par le participant «miroblog» (Lee *et al.*, 2020). L'architecture proposée par ce dernier comprend un classificateur composé de BERT suivi d'un BiLSTM et d'un NeXtVLAD . De plus, pour les données non étiquetées , ils ont utilisé un système basé sur le niveau de confiance associé à la prédiction d'une phrase issu de BERT. Ainsi, ils arrivent à des scores de précision de 0.932, de rappel de 0.936 et F1 de 0.931 sur des corpus issus de Twitter. De ce fait, ils dépassent le score F1 du deuxième meilleur modèle de 8,4%.

Malgré cette performance intéressante, comme défis supplémentaires découlant de ce type de tâche,Ghosh *et al.* (2020) soulignent des éléments qui font écho aux sections précédentes de ce travail :

«*However, we still notice that instances with subtle humor or positive sentiment are missed by the best-performing models even if they are pretrained on a very large-scale corpora*».

5 Objectifs futurs

L'objectif de ce travail était de présenter les différents défis propres à la réalisation d'algorithmes de détection automatique de l'ironie. Dans des travaux futurs, il serait intéressant de vérifier empiriquement la dynamique des limitations théoriques présentées plus haut et de déterminer quels types d'ironies verbales semblent les plus difficilement détectables par les modèles computationnels existants. Pour ce faire, il est nécessaire d'avoir une définition concrète de ce type de discours. S'il existe peu de consensus dans la littérature linguistique à ce sujet, Beals (1995) propose la définition suivante qui semble la plus englobante des différents types d'ironie existants :

> Ironie verbale : l'utilisation d'une expression verbale pour faire semblant que quelque chose est vrai tout en soulignant quelque chose d'extrêmement faux.

« L'utilisation d'une expression verbale » fait référence à la proposition théorique de Wilson & Sperber (1992). Pour (Beals, 1995) un énoncé ironique n'est pas nécessairement qu'une mention, mais peut aussi être un usage direct d'une expression. Par « pour faire semblant que quelque chose est vrai », Beals (1995) critique la théorie du faire semblant proposé par Clark & Gerrig (1984). Selon elle, affirmer qu'un locuteur, en ironisant, jouerait un rôle s'avère trop large et non représentatif de sa relation avec l'interlocuteur. Cette description se rapproche plus de celle d'une caricature que de la production d'un énoncé ironique. Elle maintient néanmoins que le locuteur ne croit pas le propos qu'il énonce. Par « souligner quelque chose d'extrêmement faux », Beals (1995) propose de regrouper les énoncés qui sont inappropriés, non pertinents et non véridiques sous cette catégorie. L'adverbe « extrêmement » souligne le rapport souvent humoristique de l'ironie verbale.

À partir de cette définition, elle est en mesure de proposer une trentaine de types d'ironie verbale. Parmi celles-ci, on retrouve des cas classiques d'opposition de sens faisant écho à la perspective gricéenne mentionnée plus haut. Toutefois, on retrouve aussi des cas d'ironie verbale plus subtils. Par exemple, en (8), l'ironie prend forme dans une fausse causalité négligeant volontairement la cause réelle. Cet énoncé est ironique si on sait que le locuteur qui émet cette phrase est quelqu'un d'infâme dans ses relations de couple et que son travail ne joue aucun rôle dans sa situation matrimoniale. On retrouve aussi des cas où l'ironie peut prendre naissance dans le choix des mots utilisés. En (9), c'est le fait de désigner une librairie comme étant des vendeurs de drogue qui est ironique. De plus, Beals (1995) mentionne des types d'ironie qui se manifestent sous forme de question où la réponse est évidente, comme en (10).

7. Je me retrouve célibataire parce que je travaille beaucoup trop.
8. Depuis que je suis sobre, je vais à la libraire du Square pour avoir mon fix.
9. Qui voudrait réellement avoir des conditions de vie décentes en ayant accès à un salaire minimum adéquat ?

Par ailleurs, comme le souligne Beals (1995), l'ironie ne s'exprime pas que verbalement. On retrouve des situations ironiques, comme en (11), qui découlent de la mise en relation de deux événements distincts.

10. Le patron de segway est mort en conduisant son segway en bas d'une falaise.

Avec une définition claire, faisant idéalement consensus, et une catégorisation exhaustive de l'ironie verbale, il est possible de déterminer quels sont les types les plus difficilement identifiables par les systèmes de détection automatique. Ainsi, il est possible d'avoir de meilleurs résultats.

De plus, il sera nécessaire d'évaluer le niveau d'erreur acceptable émis par les algorithmes de détection automatique de l'ironie. Il nous est actuellement impossible de savoir à quel point ces derniers sont meilleurs ou moins bons que les humains pour effectuer cette tâche. Les corpus d'entraînement étant annotés par ces derniers, on s'attend généralement à ce que les algorithmes puissent réussir à détecter efficacement les énoncés qui sont ironiques. Néanmoins, certaines phrases peuvent apparaître beaucoup plus ambiguës que d'autres. Par exemple, (Van Hee, 2017) rapporte le tweet en (7) où, sans la présence du hashtag, on se retrouverait face à une absence d'indice permettant à nous et aux algorithmes de bien détecter la présence d'ironie.

11. There is a thing called an all nightery and apparently, I wanna pull one #not.

Il existe un ensemble de stratégie discursive pour faire comprendre à notre interlocuteur qu'un énoncé produit sera ironique. Ces dernières dépendent des modalités utilisées pour transmettre ce type de discours. Dans une conversation, à l'oral, il sera possible de faire varier sa prosodie pour bien se faire comprendre (Bryant, 2010) ou, même, de faire certaines expressions faciales particulières pour désambiguïser son propos (Deliens *et al.*, 2018). Il est aussi important de noter que nous serions moins portés à utiliser l'ironie lorsque nous connaissons mal notre interlocuteur (Gibbs, 2000). Cohn-Gordon & Bergen (manuscrit) proposent même que l'ironie soit une façon de consolider des informations partagées entre deux personnes. De ce fait, un locuteur se doit de laisser le plus de clés possible à son interlocuteur pour que ce dernier soit capable de bien décoder son propos. L'espace d'informations partagées entre ces derniers est beaucoup plus dynamique à l'oral qu'à l'écrit. Au travers de cette dernière, un locuteur pourra employer diverses stratégies aux objectifs similaires comme l'utilisation particulière d'émojis, de majuscules et/ou de ponctuations. Ces traits sont utilisés dans la plupart des algorithmes de détection (Carvalho *et al.*, 2009; Karoui, 2017) et offrent généralement de bons résultats.

6 Conclusion

Les avancées techniques au niveau de la détection automatique de l'ironie dépendent des recherches linguistiques sur ce type de phénomène. Leur progrès nécessite la prise en considération d'une caractérisation claire du phénomène et d'une adéquation entre cette dernière et les corpus d'entraînement

utilisés. En déterminant les types d'ironie les plus difficilement identifiables par les systèmes de classifications et en caractérisant leurs manifestations, il sera possible dans le futur d'adapter ces derniers en conséquence et d'augmenter leurs performances.

Références

ALBA-JUEZ L. & ATTARDO S. (2014). The evaluative palette of verbal irony. *Evaluation in context*, **242**.

ALIAS-I (2014). Lingpipe natural language toolkit.

ATTARDO S. (2000). Irony as relevant inappropriateness. *Journal of pragmatics*, **32**(6), 793–826.

BEALS K. P. (1995). *A linguistic analysis of verbal irony*. Thèse de doctorat, University of Chicago, Department of Linguistics.

BOUAZIZI M. & OHTSUKI T. (2015). Sarcasm detection in twitter :" all your products are incredibly amazing ! ! !"-are they really ? In *2015 IEEE Global Communications Conference (GLOBECOM)*, p. 1–6 : IEEE.

BRYANT G. A. (2010). Prosodic contrasts in ironic speech. *Discourse Processes*, **47**(7), 545–566.

CARVALHO P., SARMENTO L., SILVA M. J. & DE OLIVEIRA E. (2009). Clues for detecting irony in user-generated contents : oh... ! ! it's" so easy" ;-. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, p. 53–56.

CLARK H. H. & GERRIG R. J. (1984). On the pretense theory of irony.

COHN-GORDON R. & BERGEN L. (manuscrit). Verbal irony, pretense, and the common ground.

DAVIDOV D., TSUR O. & RAPPOPORT A. (2010). Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, p. 107–116.

DELIENS G., ANTONIOU K., CLIN E., OSTASHCHENKO E. & KISSINE M. (2018). Context, facial expression and prosody in irony processing. *Journal of memory and language*, **99**, 35–48.

EKE C. I., NORMAN A. A., SHUIB L. & NWEKE H. F. (2020). Sarcasm identification in textual data : systematic review, research challenges and open directions. *Artificial Intelligence Review*, **53**(6), 4215–4258.

FARÍAS D. I. H., PATTI V. & ROSSO P. (2016). Irony detection in twitter : The role of affective content. *ACM Transactions on Internet Technology (TOIT)*, **16**(3), 1–24.

GHOSH A. & VEALE T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, p. 161–169.

GHOSH D., VAJPAYEE A. & MURESAN S. (2020). A report on the 2020 sarcasm detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, p. 1–11.

GIBBS R. W. (2000). Irony in talk among friends. *Metaphor and symbol*, **15**(1-2), 5–27.

GODDARD C. (2018). "joking, kidding, teasing" : Slippery categories for cross-cultural comparison but key words for understanding anglo conversational humor. *Intercultural Pragmatics*, **15**(4), 487–514.

GRICE H. P. (1975). Logic and conversation. In *Speech acts*, p. 41–58. Brill.

JORGENSEN J. (1996). The functions of sarcastic irony in speech. *Journal of pragmatics*, **26**(5), 613–634.

JOSHI A., SHARMA V. & BHATTACHARYYA P. (2015). Harnessing context incongruity for sarcasm detection. 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. 757–762.

KAROUI J. (2017). *Détection automatique de l'ironie dans les contenus générés par les utilisateurs*. Thèse de doctorat, Université de Toulouse 3 Paul Sabatier ; Faculté des Sciences Economiques et ....

KERBRAT-ORECCHIONI C. (1978). *L'ironie*. Presses universitaires de Lyon.

KREUZ R. & CAUCCI G. (2007). Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on computational approaches to Figurative Language*, p. 1–4.

LEE H., YU Y. & KIM G. (2020). Augmenting data for sarcasm detection with unlabeled conversation context. In *Proceedings of the Second Workshop on Figurative Language Processing*, p. 12–17.

MARTINI A. T., FARRUKH M. & GE H. (2018). Recognition of ironic sentences in twitter using attention-based lstm. *International Journal of Advanced Computer Science and Applications*, **9**(8).

NILSEN E. S., GLENWRIGHT M. & HUYDER V. (2011). Children and adults understand that verbal irony interpretation depends on listener knowledge. *Journal of Cognition and Development*, **12**(3), 374–409.

NUNBERG G. (2001). *The Way We Talk Now : Commentaries on Language and Culture from NPR's" Fresh Air"*. Houghton Mifflin Harcourt.

PORIA S., CAMBRIA E., HAZARIKA D. & VIJ P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, p. 1601–1612.

PTÁČEK T., HABERNAL I. & HONG J. (2014). Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics : Technical papers*, p. 213–223.

RILOFF E., QADIR A., SURVE P., DE SILVA L., GILBERT N. & HUANG R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, p. 704–714.

STRAPPARAVA C., STOCK O. & MIHALCEA R. (2011). Computational humour. In *Emotion-oriented systems*, p. 609–634. Springer.

SULIS E., FARÍAS D. I. H., ROSSO P., PATTI V. & RUFFO G. (2016). Figurative messages and affect in twitter : Differences between# irony,# sarcasm and# not. *Knowledge-Based Systems*, **108**, 132–143.

TSUR O., DAVIDOV D. & RAPPOPORT A. (2010). Icwsm—a great catchy name : Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4.

VAN HEE C. (2017). *Can machines sense irony ? : exploring automatic irony detection on social media*. Thèse de doctorat, Ghent University.

WALKER M. A., TREE J. E. F., ANAND P., ABBOTT R. & KING J. (2012). A corpus for research on deliberation and debate. In *LREC*, volume 12, p. 812–817 : Istanbul, Turkey.

WALLACE B. C. (2015). Computational irony : A survey and new perspectives. *Artificial intelligence review*, **43**(4), 467–483.

WILSON D. & SPERBER D. (1992). On verbal irony. *Lingua*, **87**(1), 53–76.

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

État de l'art en compression multi-phrases pour la synthèse de documents

Kévin Espasa[1, 2]
(1) Syllabs, 35-37 rue Chanzy, 75011 Paris, France
(2) LS2N, 2 Chemin de la Houssinière, 44322 Nantes, France
espasa@syllabs.com, kevin.espasa@univ-nantes.fr

RÉSUMÉ

La compression multi-phrases est utilisée dans différentes tâches de résumé (microblogs, opinions, réunions ou articles de presse). Leur objectif est de proposer une reformulation compressée et grammaticalement correcte des phrases sources tout en gardant les faits principaux. Dans cet article, nous présentons l'état de l'art de la compression multi-phrases en mettant en avant les différents corpus et outils à disposition. Nous axons notre analyse principalement sur la qualité grammaticale et informative plus que sur le taux de compression.

ABSTRACT

State-of-the-art of multi-sentence compression for document summarization

Multi-sentence compression (MSC) is used in various summary tasks (microblog, opinion, meeting or news articles). The aim is to generate a grammatical and reduced compression from multiple source sentences while retains their main facts. In this article, we present the state of the art of MSC and the different corpora and tools available. We focus our analysis more on grammatical and informative quality than on compression rate.

MOTS-CLÉS : compression multi-phrases, état de l'art, jeu de données.

KEYWORDS: multi-sentence compression, state-of-the-art, datasets.

1 Introduction

La compression de phrases a pour objectif à partir d'une phrase en entrée d'en produire une nouvelle plus courte, grammaticalement correcte et tout aussi informative (Jing & McKeown, 2000). Principalement utilisées pour des tâches de résumé, ces méthodes peuvent se séparer en deux classes. Les méthodes par suppression (Filippova *et al.*, 2015; Wang *et al.*, 2017; Zhao *et al.*, 2018) cherchent à produire un résumé en supprimant les mots inutiles tandis que celles par abstraction (Choi *et al.*, 2019; Yu *et al.*, 2018) proposent une reformulation de la phrase en y ajoutant de nouveaux mots.

C'est à partir des travaux de Barzilay & McKeown (2005) que plusieurs phrases sont proposées en entrée d'un système de fusion de phrases. Le système doit permettre d'obtenir une phrase fluide et concise reflétant les faits communs à un ensemble de phrases partageant un même thème. Par la suite, Filippova & Strube (2008) proposent de ne plus se limiter aux faits communs mais d'utiliser la complémentarité des phrases partageant un même thème pour produire une phrase profitant de l'ensemble des faits. C'est à partir de Filippova (2010) que la tâche est nommée compression multi-

phrases. L'auteure considère que l'approche visant à produire une nouvelle phrase en gardant les faits importants présents dans un ensemble de phrases sources s'apparente plus à la tâche de compression de phrases qu'à une tâche de fusion de phrases.

source	Le cofondateur d'Apple nous a quitté mercredi 5 octobre à l'âge de 56 ans. Steve Jobs est mort mercredi, à la suite d'une longue maladie . Le fondateur d'Apple s'est éteint mercredi, à 56 ans, des suites d'un cancer du pancréas.
ref.	Steve Jobs, co-fondateur d'Apple, s'est éteint ce mercredi 5 octobre à 56 ans.
gen.	le co-fondateur d'apple steve jobs est mort le 5 octobre . apple steve jobs est mort le 5 octobre .

TABLE 1 – Exemple de phrases sources et de compressions issu de Boudin & Morin (2013)

La table 1 présente un exemple de phrases sources et d'une compression de référence (ref.) issu du corpus [1] Boudin & Morin (2013) ainsi que deux compressions générées automatiquement (gen.) par l'algorithme Takahe [2].

Dans cet article, nous présentons un état de l'art des méthodes de compression multi-phrases. Nous cherchons à évaluer les différentes méthodes, la facilité de reproductibilité et les différents corpus d'évaluation. Notre objectif par la suite est d'utiliser ces méthodes afin de produire une reformulation d'un ensemble de documents partageant un même thème dans un cadre industriel. Nous considérons ces approches comme pertinentes car elles permettraient de générer des textes plus ou moins compressés en fonction des besoins (de la brève à l'article détaillé).

Dans la suite de l'article, nous commençons par développer la problématique liée à la tâche en section 2. Puis nous discutons des différentes méthodes dans la section 3. La section 4 présente les différents jeux de données et les méthodes d'évaluation. La section 5 décrit les expérimentations que nous avons faites. Enfin nous concluons et présentons nos perspectives de recherches.

2 Compression multi-phrases

La compression multi-phrases cherche, à partir d'un regroupement de phrases similaires (i.e. partageant un même thème), à proposer une ou plusieurs reformulations respectant des contraintes d'information, de compression et de grammaticalité. La contrainte d'information a pour but de produire une reformulation la plus informative possible. Plus précisément, la méthode doit être capable d'identifier les faits les plus pertinents et de les restituer en sortie.

Considérons un ensemble de phrases S en entrée ayant en moyenne n termes. Afin de respecter la contrainte de compression la ou les phrases en sortie S' doivent avoir une moyenne de termes n' inférieure à n. Le respect de la grammaticalité est également important, les phrases générées doivent contenir le moins d'erreurs grammaticales.

La compression de phrases et la compression multi-phrases sont utilisées dans différentes tâches comme le résumé de microblogs (Sharifi *et al.*, 2010), d'opinions (Ganesan *et al.*, 2010), de réunions (Shang *et al.*, 2018) ou d'articles de presse (Nayeem *et al.*, 2018). Pour nos travaux, nous nous plaçons dans cette dernière tâche. Nous cherchons une méthode capable de résumer un ensemble

1. https://github.com/boudinfl/lina-msc/tree/master/src
2. https://github.com/boudinfl/takahe

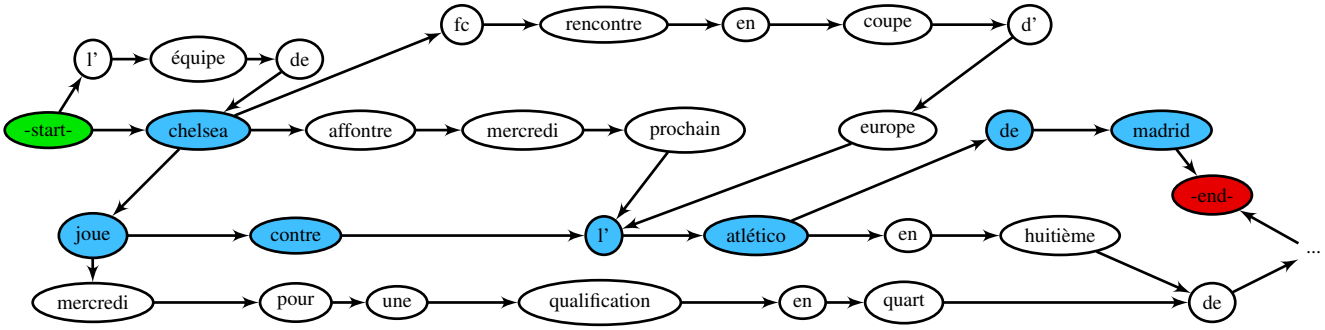


FIGURE 1 – Graphe représentant les phrases 1 à 4 et un chemin possible

de phrases pour produire une ou plusieurs phrases candidates. Étant dans un cadre industriel, nous mettons l'accent principalement sur la grammaticalité et l'informativité de la phrase. Il est également intéressant dans notre cadre de produire une reformulation des phrases sources. Plus précisément, nous souhaitons que les phrases générées possèdent le moins de mots en commun avec les sources.

3 Approches étudiées

La résolution de la tâche de compression multi-phrases utilise principalement une représentation des mots à l'aide d'un graphe (Filippova, 2010; Boudin & Morin, 2013; Linhares Pontes *et al.*, 2018; ShafieiBavani *et al.*, 2016). Les méthodes diffèrent cependant dans le regroupement entre les différents mots ainsi que lors du parcours du graphe.

3.1 Filippova (2010)

Filippova (2010) propose pour la compression multi-phrases une méthode basée sur la représentation des mots à l'aide de graphes. Un graphe $G = (N, A)$, avec N l'ensemble des nœuds et A l'ensemble des arêtes, est construit par l'ajout successif de mots des phrases d'un ensemble $S = s_1, ..., s_n$.

Les exemples de phrases suivantes servent à illustrer le fonctionnement de la méthode.

1. Chelsea joue contre l'Atlético en huitième de *finale de la Ligue des Champions*
2. Chelsea affontre mercredi prochain l'Atlético de Madrid
3. Chelsea FC rencontre en coupe d'Europe l'Atlético de Madrid
4. L'équipe de Chelsea joue mercredi pour une qualification en quart de *finale*

La Figure 1 représente le graphe construit en utilisant l'algorithme. En vert, le nœud de début et en rouge celui de fin de parcours. Les nœuds en bleu représentent un chemin possible lors du parcours. Afin d'améliorer la lisibilité du graphe, les parties en italique des phrases 1 et 4 ont été remplacées par des points de suspensions dans la Figure 1.

Chaque mot de la première phrase s_1 (la ponctuation étant exclue) est transformé en nœud n. Puis pour chaque phrase suivante, les mots sont ajoutés au graphe dans l'ordre suivant :

1. les mots grammaticaux n'ayant pas de nœuds candidats au regroupement ou pas d'ambiguïté possible sur le candidat,

2. les mot grammaticaux ayant plusieurs candidats au regroupement possible,
3. les mots vides.

Pour les mots du premier groupe, un mot est regroupé avec un nœud existant si ils sont similaires et ont la même étiquette morphosyntaxique et qu'aucun mot de la phrase s n'a déjà été regroupé avec le nœud n du graphe. Dans le cas où le mot ne peut être regroupé, un nouveau nœud est ajouté à G.

Pour les deux derniers groupes, en cas d'ambiguïté les mots suivant et précédent de chaque candidat sont comparés pour choisir le meilleur regroupement. Les mots vides sont regroupés seulement si leur contexte immédiat est similaire, sinon un nouveau nœud est créé. Le premier mot de chaque phrase est connecté avec un nœud de départ (-*start*- dans la Figure 1) tandis que le dernier est connecté avec un nœud de fin (-*end*- dans la Figure 1). Les nœuds sont reliés par des arêtes unidirectionnelles suivant leur ordre dans la phrase et un poids par défaut de 1 est ajouté aux arêtes.

Une fois le graphe obtenu, le poids de chaque arête est calculé en utilisant l'équation 1. $freq(i)$ et $freq(j)$ représentent respectivement la fréquence du mot i et du mot j. La fonction de cohésion (équation 2) calcule pour chaque i et j leur fréquence divisée par la distance entre les mots dans chaque phrase. Le but étant de privilégier les mots qui apparaissent le plus souvent ensemble.

Par la suite un algorithme de K-plus court chemin est utilisé pour parcourir le graphe. Le parcours vise à trouver un chemin de taille définie (8 dans l'article) tout en minimisant la somme des arêtes parcourues. Afin d'obtenir une phrase grammaticalement correcte en sortie, un nœud contenant un verbe doit être traversé. Les scores sont finalement normalisés en fonction de la taille de phrase générée puis réordonnés. Le chemin ayant le plus petit poids est alors la meilleure compression.

$$w(i,j) = \frac{cohesion(i,j)}{freq(i) \times freq(j)} \tag{1}$$

$$cohesion(i,j) = \frac{freq(i) + freq(j)}{\sum_{s \epsilon S} distance(s,i,j)^{-1}} \tag{2}$$

La méthode offre l'avantage de n'être dépendante que d'un outil d'étiquetage morphosyntaxique et une liste de mots outils. Pour l'anglais, l'auteure génère une liste de 600 mots vides spécifiques aux articles d'actualités pour l'anglais incluant certains verbes (*said*, *seems*) ainsi qu'une liste publique de 180 mots vides [3] pour l'espagnol. La liste générée pour l'anglais n'est cependant pas mise à disposition et la façon dont elle est créée n'est pas décrite pour reproduire les expériences.

3.2 Boudin & Morin (2013)

Boudin & Morin (2013) proposent une amélioration de la méthode de Filippova (2010) en y incluant la ponctuation ainsi qu'en utilisant une méthode d'extraction de termes clefs pour le calcul des poids. Les auteurs reprennent les trois étapes successives d'ajout des mots et en ajoutent une quatrième pour la ponctuation. En cas d'ambiguïté lors de l'ajout, le contexte immédiat (mot suivant et mot précédent) sont comparés.

Dans les résultats donnés par Filippova, l'information est restituée en totalité dans 52 % des cas en anglais (40 % en espagnol). Afin d'améliorer la conservation de l'information, une méthode pour réordonner les phrases générées en fonction des termes clefs qu'elles contiennent est mise en place.

3. https://www.ranks.nl/stopwords/spanish

La méthode se déroule en deux étapes. Tout d'abord, un graphe pondéré est construit pour chaque regroupement de phrases. Chaque nœud contient le mot et son étiquette morphosyntaxique. Les arêtes sont pondérées en utilisant la cooccurrence entre les mots présents dans des nœuds. Un poids d'importance du nœud est calculé (équation 3) en utilisant la méthode TextRank (Mihalcea & Tarau, 2004).

$$TextRank(A_i) = (1-d) + d \times \sum_{V_j \epsilon adj(V_i)} \frac{w_{ji}}{\sum_{V_k \epsilon adj(V_j)} w_{kj}} S(A_j) \tag{3}$$

Le score du nœud A_i est calculé à l'aide de l'équation 1, de A_j qui représente les nœuds en lien direct avec V_i et le facteur d défini à 0,85. La seconde étape consiste à extraire, pour chaque phrase générée, les expressions clefs ayant la forme suivante : $(ADJ) * (NPP|NC) + (ADJ)*$. Une fois extrait, le score d'une expression clef k est calculée avec l'équation 4. Puis le score général de la phrase est défini en utilisant la somme des scores des poids du chemin pour construire la phrase c divisée par sa longueur et par la somme des scores des expressions clefs (l'équation 5).

$$score(k) = \frac{\sum_{w \epsilon k} TexTrank(w)}{|k| + 1} \tag{4}$$

$$score(c) = \frac{\sum_{i,j \epsilon path(c)} w_{i,j}}{|c| + \sum_{k \epsilon c} score(k)} \tag{5}$$

D'après les résultats présents dans Boudin & Morin (2013), cette méthode permet une restitution totale de l'information de 62,5 % des cas contre 43,3 % pour Filippova (2010) sur un corpus français. À noter, qu'une différence de 8,7 % est présente dans la restitution totale de l'information entre l'approche de Filippova (2010) sur l'anglais et la reproduction de la méthode sur le français par Boudin & Morin (2013). Outre la langue, il est possible comme pour la comparaison entre l'espagnol et l'anglais que la taille de la liste des mots vides, 203 mots pour le français contre 600 pour Filippova (2010) avec l'anglais, ait un impact sur les performances. Notons également que l'amélioration de l'information entraîne une diminution 7,5 % de la grammaticalité des phrases générées.

Les auteurs ont mis en ligne le code [4] de leur algorithme ainsi qu'une implémentation de celui de Filippova (2010). Les listes de mots vides utilisées sont également disponibles.

3.3 ShafieiBavani *et al.* (2016)

En reprenant les travaux de Filippova (2010) et Boudin & Morin (2013), ShafieiBavani *et al.* (2016) proposent une méthode utilisant la détection d'expressions polylexicales et le remplacement de synonymes afin d'améliorer la grammaticalité et l'information dans les phrases générées. La méthode utilise l'approche de Filippova (2010) pour la construction d'un graphe à partir d'un ensemble de phrases. Deux modifications sont apportées afin de prendre en compte les expressions polylexicales et les synonymes.

La première consiste à détecter les expressions polylexicales avec l'outil jMWE (Kulkarni & Finlayson, 2011) puis à les remplacer par un synonyme composé d'un mot à l'aide de Wordnet. Par exemple

4. https://github.com/boudinfl/takahe

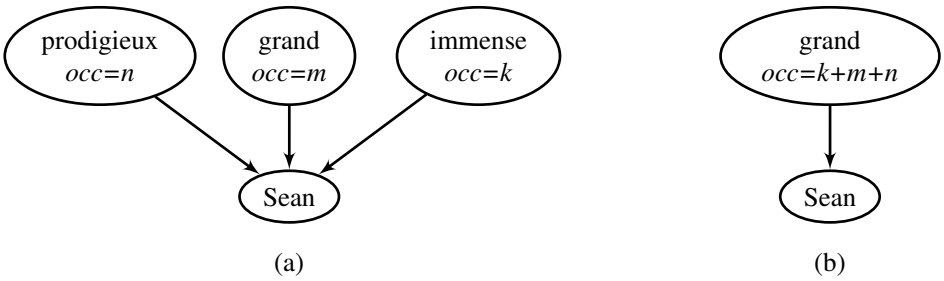


FIGURE 2 – Graphe représentant le regroupement de synonymes

l'expression *passer l'arme à gauche* dans *Sean Connery vient de passer l'arme à gauche* sera détectée comme une expression polylexicale puis convertie en *mourir* lors de son ajout dans le graphe.

La seconde modification consiste à regrouper les synonymes dans un même nœud afin de : (i) réduire l'ambiguïté lors du regroupement entre un mot et un nœud candidat et (ii) limiter le nombre de chemins et augmenter le score de cooccurrences entre un mot et un ensemble de mots synonymes. Pour les phrases suivantes, lors de la construction du graphe de la Figure 2 *grand*, *immense* et *prodigieux* seront regroupés dans un même nœud. Dans notre exemple, k, n et m ont un score de un (a), le score de cooccurrences total sera alors de trois pour le nœud *immense* (b) :

1. Le grand Sean Connery est mort
2. L'immense Sean Connery est mort
3. Le prodigieux Sean Connery est mort

Une fois le graphe pondéré créé, un parcours à l'aide de l'algorithme des K-plus courts chemins est effectué. Puis un algorithme pour réordonner la liste de candidats est utilisé. Les auteurs utilisent en plus de la méthode d'équation de Boudin & Morin (2013), un modèle de langue créé à partir d'étiquettes morphosyntaxiques afin d'augmenter la validité syntaxique de la phrase (équation 6). Le score final d'une phrase c est représenté par un facteur μ, le score calculé par l'équation 4 et le score du modèle de langue (équation 7).

$$score_{LM}(c) = 10^{\frac{logprob(c)}{|c|}} \tag{6}$$

$$score_{final}(c) = \mu \times score(c) + (1 - \mu) \times score_{LM}(c) \tag{7}$$

La méthode permet grâce à la création d'un graphe ayant moins de chemins possibles une amélioration de la compression et de la grammaticalité. Cependant, le remplacement des expressions polylexicales et de certains termes par leur synonyme limite la variabilité des phrases générées.

3.4 Linhares Pontes *et al.* (2020)

Linhares Pontes *et al.* (2020), eux, utilisent un graphe de nœuds étiquetés pour la compression de phrases combiné à un modèle d'optimisation linéaire en nombre entiers (OLNE). Un graphe $G = (N, A)$ est construit en utilisant la méthode d'ajout successif de mots de Filippova (2010). Chaque nœud N se voit associé d'un label compris dans $K = 0, ..., |K|$. L'objectif est de parcourir le graphe en passant dans le plus de nœud étiqueté tout en ne les traversant pas plus d'une fois. Dans le

contexte de l'article, le label est associé à un mot clef, si le nœud ne contient pas de mot clef le label sera zéro.

Trois méthodes sont mises en place pour la détection des mots clefs : une indexation sémantique latente (LSI), une allocation de Dirichlet latente (LDA) et l'algorithme TextRank. Des expériences sont faites en sélectionnant les 5 ou 10 mots clefs les plus pertinents de chaque méthode. Les mots clefs extraits à l'aide de la méthode LDA sont les plus présents dans les corpus français, portugais et espagnol de référence. Le LDA utilisant un seuil de 10 est sélectionné.

OLNE permet de définir une fonction d'optimisation ainsi que des contraintes lors du parcours du graphe. L'objectif est définie par l'équation 8 avec $x_{i,j}$ qui représente l'existence d'un arc entre les nœuds i,j, $w_{i,j}$ est l'équation 1 et b_k indique la présence du mot clef k dans la solution.

$$score_{opt}(s) = Minimize(\sum_{(i,j)\epsilon A} w_{i,j} \cdot x_{i,j} - c \cdot \sum_{k\epsilon K} b_k) \tag{8}$$

La liste de contraintes comporte : un minimum et un maximum dans la longueur de la phrase générée, la phrase doit contenir des mots clefs et le chemin ne doit pas traverser plusieurs fois le même nœud. Afin de prendre en compte la taille des phrases générées, le score est normalisé (équation 9).

$$score_{norm}(s) = \frac{e^{score_{opt}(s)}}{|c|} \tag{9}$$

3.5 Zhao *et al.* (2019)

Enfin, Zhao *et al.* (2019) proposent une méthode basée sur un bi-LSTM pour réécrire les compressions générées. Un corpus en phrases similaires est construit (voir section 4.1). La construction permet d'obtenir un corpus A de 140 000 regroupements de phrases composés chacun de 2 à 4 phrases.

La méthode de création du modèle se divise en 3 étapes. La première consiste à partir du corpus A de créer à l'aide de la méthode de Boudin & Morin (2013) un corpus de compression nommé B. Les expressions polylexicales, les verbes, les adjectifs et les noms compris dans chaque phrase du corpus B sont remplacés par leur plus petit synonyme possible à l'aide de Wordnet [5] et de PPDB 2.0 [6]. Le but étant d'obtenir un troisième corpus nommé C avec une compression maximale.

Lors de la seconde étape, un modèle bi-LSTM encodeur décodeur est entraîné avec en entrée le C et en sortie B. Ce dernier permet de générer à partir d'un corpus C' contenant 1 millions de phrases (le nombre de tokens moyen de C' équivaut à celui de C) un corpus B'. La dernière étape consiste à entraîner un modèle bi-LSTM avec en entrée les corpus B et B' et en sortie les corpus C et C'. L'objectif de ce modèle est de reformuler les compressions générées par d'autres méthodes en améliorant la grammaticalité et en y ajoutant de nouveaux mots (grâce à l'étape 2).

L'approche de Zhao *et al.* (2019) a pour but de produire une réécriture plus compressée et introduisant des mots non présents dans la compression d'origine. La réécriture reste cependant dépendante de l'approche de compression utilisée, la propagation d'erreurs informatives n'est pas à négliger.

5. https://wordnet.princeton.edu
6. http://paraphrase.org

3.6 Synthèse

Les méthodes utilisant des graphes proposent une compression des faits présents dans plusieurs phrases. Les différentes façons dont les scores sont calculés présentent l'avantage de ne pas pénaliser un regroupement de phrases légèrement bruité. En effet, l'information restituée en première sera celle présente le plus souvent. Ce qui est intéressant notamment lorsque le regroupement de phrases se fait de manière automatique. Nous nous plaçons dans une problématique de synthèse d'articles de presse, les méthodes de Filippova (2010); Boudin & Morin (2013); Linhares Pontes *et al.* (2018) ont le désavantage de ne pas ajouter de variantes lexicales aux phrases en sortie du système. Les mots en sortie correspondent obligatoirement aux entrées. Zhao *et al.* (2019); ShafieiBavani *et al.* (2016) tentent de proposer des méthodes pour remplacer les mots ou regrouper les synonymes. ShafieiBavani *et al.* (2016) s'appuient sur un outil d'extraction de mots polylexicaux seulement disponible en anglais et Zhao *et al.* (2019) ne précisent pas la façon dont les expressions polylexicales sont détectées. Le remplacement des expressions polylexicales n'est cependant pas sans risque, l'ambiguïté existe et n'est pas facilement détectable.

4 Jeux de données et mesures d'évaluation

Dans les travaux cités précédemment, les langues utilisées sont l'anglais, l'espagnol, le français, et le portugais. Des jeux d'évaluation ont été mis en ligne pour l'espagnol, le français et le portugais, mais il n'existe aucun jeu de référence, à notre connaissance, pour l'anglais. Notons quand même qu'une procédure standard de création des corpus d'évaluation existe. Nous décrirons ensuite les différentes mesures d'évaluation mises en œuvre dans le cadre de la compression multi-phrases.

4.1 Méthodologie de création de jeux de données

La méthode de Filippova (2010) consiste à collecter des regroupements d'articles de presse traitant d'un même événement à l'aide de Google News [7]. Ce dernier présente l'avantage d'avoir une classification et un regroupement d'articles à disposition. Les regroupements, manuellement extraits, contiennent plusieurs articles, entre 10 et 30 pour Filippova (2010) et au moins 20 pour Boudin & Morin (2013). La première phrase de chaque article est conservée (sauf en cas de duplicata où la phrase est retirée). Elle est considérée comme étant un bon résumé de l'article et est utilisée en référence dans la tâche de résumé (Dang, 2005).

Boudin & Morin (2013) ajoutent une compression manuelle pour le jeu de référence. La compression de référence consiste à demander à des locuteurs natifs de produire, à l'aide des phrases sources, la meilleure compression possible en utilisant le moins de nouveaux mots possibles.

Zhao *et al.* (2019) proposent une méthode de construction automatique d'un corpus de phrases similaires. Les auteurs appliquent une méthode de similarité des bigrammes sur le corpus English Gigaword. Une limite basse et une limite haute sont ajoutées afin d'éviter un regroupement de phrases pas assez ou trop similaires. Une évaluation humaine sur cinquante regroupements de phrases donne un résultat de 90 % de regroupement correct (Zhao *et al.*, 2019). Le corpus ainsi obtenu contient 140 572 groupements de phrases contenant entre 2 et 4 phrases chacun. Les auteurs créent un corpus

7. https://news.google.com

de référence en sélectionnant aléatoirement 150 phrases et en demandant à deux locuteurs natifs d'en produire une compression.

Article	Langue	Corpus	Disponible	Licence
(Filippova, 2010)	anglais	Google News	non	
	espagnol	Google News	non	
(Boudin & Morin, 2013)	français	Google News	oui [8]	MIT
(ShafieiBavani *et al.*, 2016)	anglais	Google News	non	
(Linhares Pontes *et al.*, 2020)	espagnol	Google News	oui [9]	GPL
	portugais	Google News	oui [10]	GPL
(Zhao *et al.*, 2019)	anglais	English Gigaword	oui [11]	LDC
	anglais	Fusion Corpus		

TABLE 2 – Caractéristiques des données utilisées dans les différents articles cités

La table 2 présente les caractéristiques des données utilisées dans les articles. Majoritairement, les auteurs utilisent Google News pour créer leur jeu de données mais seulement trois d'entre eux sont disponibles gratuitement. Notons qu'aucun corpus de référence n'existe pour l'anglais à ce jour et que le Fusion Corpus (McKeown *et al.*, 2010) utilisé comme second corpus d'évaluation par Zhao *et al.* (2019) n'est plus disponible à l'adresse indiqué dans leur article.

4.2 Description des corpus disponibles

Nous décrirons dans cette partie les corpus disponibles pour le français, le portugais et l'espagnol. Boudin & Morin (2013) ont mis à disposition un corpus libre en français contenant 618 phrases ainsi que les 120 phrases de références associées aux 40 clusters (3 phrases de références par cluster). Les longueurs moyennes en nombre de tokens pour la source et la référence sont respectivement de 32,7 et de 19,7. Ce qui implique un taux de compression manuel de 60 %.

Des corpus en espagnol et en portugais ont été créés par Linhares Pontes *et al.* (2020). Ces derniers sont également libres. Le corpus portugais contient 40 clusters composés de 544 phrases sources et 80 phrases de références. Le corpus espagnol se compose 800 phrases réparties dans 40 clusters et 4 phrases de références par cluster. Le taux de compression entre la source et la référence est en moyenne de 54 % pour le portugais et 61 % pour l'espagnol.

La table 3 récapitule les différentes informations sur les corpus : nombre de phrases, nombre minimum et maximum de phrases par cluster, nombre minimum, maximum et moyen de tokens par phrases ainsi que le taux de compression entre les phrases de références et les phrases sources.

4.3 Mesures d'évaluation

Plusieurs types de mesures existent pour évaluer la qualité d'une compression multi-phrases, ces mesures peuvent être séparées en deux familles : les automatiques et les manuelles. Les méthodes

8. https://github.com/boudinfl/lina-msc
9. http://juanmanuel.torres.free.fr/corpus/msf2/publications.html
10. http://juanmanuel.torres.free.fr/corpus/msf2/publications.html
11. https://catalog.ldc.upenn.edu/LDC2011T07

	(Boudin & Morin, 2013)		(Linhares Pontes *et al.*, 2020)			
Langue	Français		Portugais		Espagnol	
#clusters	40		40		40	
Type	**Source**	**Référence**	**Source**	**Référence**	**Source**	**Référence**
#phrases	618	120	544	80	800	160
min. phrases	7	3	9	2	20	4
max. phrases	36	3	22	2	20	4
#tokens	20 225	2 362	17 998	1 426	30 589	3 695
min. tokens	10		11	10	16	16
max. tokens	82		77	26	100	35
moy. tokens	32,7	19,7	33,1	17,8	38,2	23,1
taux de compression		60 %		54 %		61 %

TABLE 3 – Caractéristiques des corpus

automatiques d'évaluation sont devenues courantes dans les tâches d'évaluation des textes générés que ce soit pour la traduction automatique ou le résumé. Dans le cas de la compression multi-phrases, différentes mesures sont utilisées : BLEU, ROUGE ou encore METEOR afin de comparer la similarité entre les phrases de références et les phrases générées. Cependant, ces méthodes ne permettent pas de comparer la qualité grammaticale et la quantité d'informations pertinentes restituées.

Les méthodes manuelles pour la compression multi-phrases cherchent à évaluer ces aspects (cf. table 4. Barzilay & McKeown (2005) proposent une méthode de notation de la qualité grammaticale des phrases générées : *parfait* si la phrase est grammaticalement correcte (2 points), *presque* si la phrase ne requiert qu'une correction minime : une seule erreur (1 point) et *agrammaticale* si la phrase est incorrecte (0 point). Filippova (2010) reprend ce système pour noter cette fois la qualité de l'information restituée : *n/a* si le regroupement de phrases est trop bruité et ne peut donc pas produire de synthèse, *parfait* si la phrase contient les informations du thème principal (2 points), *en relation* si phrase contient une partie des informations du thème (1 point) et *sans relation* si la phrase n'a pas de rapport avec le thème.

Caractéristique	**Description**	**Point(s)**		
		2	1	0
Grammaticalité	grammaticalité parfaite	×		
	correction minime		×	
	agrammaticale			×
Information	parfaitement restituée	×		
	quasiment restituée		×	
	non restituée			×

TABLE 4 – Notation de l'information et de la grammaticalité

Le taux de compression est également évalué. Cela consiste à diviser le nombre de tokens de la phrase générée par le nombre de tokens moyens des phrases en entrée du système.

La table 5 récapitule les méthodes utilisées dans les différents articles cités. Les scores de grammaticalité, d'information et le taux de compression sont présents à chaque fois, ce qui montre leur importance dans la compression de phrase. Nous pouvons noter que Filippova (2010) n'utilise pas

d'évaluation automatique. Cela s'explique par le fait que son approche est évaluée sur une sous-partie de son corpus source et donc qu'il n'y a pas de corpus de référence construit manuellement.

Articles	Gram.	Inf.	Taux comp.	BLEU	ROUGE	METEOR
(Filippova, 2010)	×	×	×			
(Boudin & Morin, 2013)	×	×	×	×	×	
(ShafieiBavani *et al.*, 2016)	×	×	×	×	×	
(Linhares Pontes *et al.*, 2018)	×	×	×	×		
(Zhao *et al.*, 2019)	×	×	×			×

TABLE 5 – La liste des métriques utilisées dans les différents articles

5 Expérimentation des méthodes

Dans cette partie, nous décrivons les expériences que nous avons réalisées. Notre objectif pour commencer était de regarder le résultat des systèmes de compression multi-phrases sur des titres d'articles de presse. Le choix de se limiter aux titres se justifie par différentes raisons : dans le cadre de nos travaux sur la synthèse de documents, il est intéressant pour nous de pouvoir reformuler un titre. L'une des complexités des méthodes de compression multi-phrases est l'alignement automatique des phrases de différents documents regroupés entre eux par similarité sémantique. L'utilisation des titres permet, dans un premier temps, de tester les approches en utilisant un corpus de phrases peu bruité. Enfin, les titres présents dans un regroupement de documents ont pour avantage d'être relativement court et de traiter du même thème.

Nous avons expérimenté les approches de Filippova (2010) et Boudin & Morin (2013) en utilisant le code mis à disposition [12] par Boudin & Morin (2013). Les deux approches sont implémentées, même s'il est à noter que la ponctuation est présente pour le système de Filippova (2010). Le code est fonctionnel en l'état mais des modifications ont été apportées pour l'intégrer dans notre processus automatique de traitement [13]. La compression sur des titres donne de bons résultats que ce soit sur des regroupements avec peu de phrases (4 ou 5 phrases) ou sur des phrases bruitées (nom du média présent dans le titre par exemple). Nous pouvons mettre en avant deux problématiques rencontrées.

La première est la gestion des entités nommées dans la restitution de l'information. Les méthodes, actuellement, ne prennent pas en compte les entités nommées dans la création du graphe, ce qui peut produire un manque d'information ou une confusion lors de la restitution. Le premier cas apparaît notamment avec les entités nommées de type fonction. Par exemple :

— Le président de Nikola a démissionné
— Nikola : le président a démissionné

Le système produira *le président a démissionné*. Cette dernière phrase est correcte mais en l'état le titre n'est pas exploitable si nous souhaitons privilégier l'informativité à la compression. Il sera ici intéressant de regrouper dans le même nœud du graphe les mots : *le président de Nikola*.

Un autre cas apparaît lorsque deux entités nommées dans une même phrase comportent un mot en commun. Par exemple avec *le groupe Renault* et *la Renault Clio*, il est important de dissocier les deux

12. https://github.com/boudinfl/takahe
13. Nous avons mis à jour le code pour qu'il puisse fonctionner en Python3 et modifié les patrons d'extractions pour qu'il fonctionne avec les étiquettes morpho-syntaxiques produites par un modèle https://spacy.io/

mots *Renault* afin de ne pas avoir en sortie : le *Groupe Renault Clio*.

La seconde problématique, plus complexe, est la capacité de l'approche à restituer une information lorsque le sujet et l'objet sont réversibles. Prenons l'exemple d'un match nul entre deux équipes de football :

— Le PSG version Mauricio Pochettino débute par un nul à Saint-Etienne
— L1 : l' ASSE décroche le nul 1 - 1 face au PSG

L'une des propositions des systèmes, de part l'utilisation des chemins du graphe, produira comme phrase : *l' asse décroche le nul à saint-etienne*. La phrase est grammaticalement correcte mais la production d'un titre comme ce dernier perd en information.

Nous avons essayé de reproduire l'expérience de Zhao *et al.* (2018) en utilisant le code qu'il a mis à disposition [14]. Le code peut se diviser en deux grands objectifs : le premier, la création du corpus parallèle, et en deuxième, la méthode de création du modèle. Le code mis à disposition n'est pas fonctionnel en l'état mais il est facilement révisable. Pour la création du modèle, les étapes 1 et 2 de son approche ne sont pas présentes. Pour l'étape 1, la création du corpus B avec l'outil de Boudin & Morin (2013) est facile à mettre en place. Il est plus compliqué de passer du corpus B au corpus C, les auteurs ne précisant pas dans l'article la façon dont les expressions polylexicales sont extraites et rien dans le code ne le fait. Pour l'étape 2, les auteurs génèrent un million de phrases compressées B' à partir d'un million de phrases C'. Nous regretterons que le modèle ne soit pas disponible ni que l'origine des phrases C' soit donnée. Pour l'étape 3, le code est disponible mais ne fonctionne pas et le modèle entraîné n'est pas disponible. Nous avons tenté de le corriger mais le manque d'informations sur certaines parties comme ce qui sert à l'entraînement du Word2Vec a rendu la tâche impossible.

6 Conclusion et travaux futurs

Dans cet article, nous avons présenté les principales méthodes de compressions de phrases, les corpus utilisés et les différentes méthodes d'évaluation. Pour nos besoins, nous avons mis l'accent sur la qualité grammaticale et de l'information restituée. Le taux de compression est secondaire par rapport aux deux autres caractéristiques.

Les articles reposent tous sur l'utilisation d'un graphe pour représenter les phrases et pour les parcourir. Les différentes améliorations apportées depuis les travaux de Filippova (2010) concernent la façon dont mettre en avant certains termes et la manière d'associer un score à une phrase générée. Il est également intéressant de souligner que les méthodes n'ont besoin que de peu d'apport extérieur pour fonctionner correctement. Une liste de mots vides et un outil d'étiquetage morpho-syntaxique sont souvent suffisants.

Les expérimentations nous ont montré que nos objectifs étaient légèrement différents de ce qui se fait dans le domaine de la compression multi-phrases. Nous cherchons à obtenir avant tout une phrase grammaticalement correcte et informative. Nous voulons également être capable d'ajouter dans la phrase générée des mots non présents dans phrases sources.

Nos travaux futurs s'intéresseront à une meilleure prise en compte des entités nommées dans les phrases afin d'obtenir une restitution plus informative tout en ne délaissant pas la qualité grammaticale de la phrase générée. Nous chercherons également à développer une méthode ajoutant de la variété lexicale en prenant en compte les difficultés sur l'ambiguïté de certains mots.

14. https://github.com/code4ai

Références

BARZILAY R. & MCKEOWN K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, **31**(3), 297–328.

BOUDIN F. & MORIN E. (2013). Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'2013)*, p. 298–305, Atlanta, GA, USA.

CHOI S. J., JUNG I., PARK S. & PARK S.-B. (2019). Abstractive sentence compression with event attention. *Applied Sciences*, **9**(19).

DANG H. T. (2005). Overview of duc 2005. In *Proceedings of the Document Understanding Conference (DUC'2005)*, p. 1–12, Vancouver, Canada.

FILIPPOVA K. (2010). Multi-sentence compression : Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, p. 322–330, Beijing, China.

FILIPPOVA K., ALFONSECA E., COLMENARES C. A., KAISER L. & VINYALS O. (2015). Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'2015)*, p. 360–368, Lisbon, Portugal.

FILIPPOVA K. & STRUBE M. (2008). Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP'2008)*, p. 177–185, Honolulu, HI, USA.

GANESAN K., ZHAI C. & HAN J. (2010). Opinosis : A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'2010)*, p. 340–348, Beijing, China.

JING H. & MCKEOWN K. R. (2000). Cut and paste based text summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL'2000)*, p. 178–185, Seattle, WA, USA.

KULKARNI N. & FINLAYSON M. (2011). jMWE : A Java toolkit for detecting multi-word expressions. In *Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World (MWE'2011)*, p. 122–124, Portland, OR, USA.

LINHARES PONTES E., HUET S., TORRES-MORENO J.-M., GOUVEIA T. & LINHARES A. (2020). A multilingual study of multi-sentence compression using word vertex-labeled graphs and integer linear programming. *Computación y Sistemas*, **24**.

LINHARES PONTES E., TORRES-MORENO J.-M., HUET S. & LINHARES A. C. (2018). A new annotated Portuguese/Spanish corpus for the multi-sentence compression task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'2018)*, p. 3192–3196, Miyazaki, Japan.

MCKEOWN K., ROSENTHAL S., THADANI K. & MOORE C. (2010). Time-efficient creation of an accurate sentence fusion corpus. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL'2010)*, p. 317–320, Los Angeles, CA, USA.

MIHALCEA R. & TARAU P. (2004). TextRank : Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'2004)*, p. 404–411, Barcelona, Spain.

NAYEEM M. T., FUAD T. A. & CHALI Y. (2018). Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'2018)*, p. 1191–1204, Santa Fe, NM, USA.

SHAFIEIBAVANI E., EBRAHIMI M., WONG R. K. & CHEN F. (2016). An efficient approach for multi-sentence compression. In R. J. DURRANT & K.-E. KIM, Éds., *Proceedings of The 8th Asian Conference on Machine Learning (ACML'2016)*, p. 414–429, Hamilton, New Zealand.

SHANG G., DING W., ZHANG Z., TIXIER A., MELADIANOS P., VAZIRGIANNIS M. & LORRÉ J.-P. (2018). Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'2018)*, p. 664–674, Melbourne, Australia.

SHARIFI B., HUTTON M.-A. & KALITA J. (2010). Summarizing microblogs automatically. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL'2010)*, p. 685–688, Los Angeles, CA, USA.

WANG L., JIANG J., CHIEU H. L., ONG C. H., SONG D. & LIAO L. (2017). Can syntax help ? improving an LSTM-based sentence compression model for new domains. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'2017)*, p. 1385–1393, Vancouver, Canada.

YU N., ZHANG J., HUANG M. & ZHU X. (2018). An operation network for abstractive sentence compression. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'2018)*, p. 1065–1076, Santa Fe, NM, USA.

ZHAO Y., LUO Z. & AIZAWA A. (2018). A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'2018)*, p. 170–175, Melbourne, Australia.

ZHAO Y., SHEN X., BI W. & AIZAWA A. (2019). Unsupervised rewriter for multi-sentence compression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'2019)*, p. 2235–2240, Florence, Italy.

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

Modification d'un modèle de liage d'entités nommées end-to-end par l'ajout d'embeddings contextuels

Valentin Carpentier[1]
(1) CNRS, LIMSI, Université Paris-Saclay, 91400, Orsay, France
valentin.carpentier@limsi.fr

RÉSUMÉ

Cet article présente les expériences effectuées sur un système de liage d'entités nommées. Cette tâche se découpe en deux principales parties que sont la détection de mentions méritant d'être liées à la base de connaissance et la désambiguïsation qui permet de sélectionner l'entité finale à lier à chaque mention. Deux approches existent pour résoudre cette tâche. Il y a celle de désambiguïsation seule et celle *end-to-end* qui effectue les deux sous-tâches simultanément. Nous nous sommes intéressés au modèle *end-to-end* atteignant l'état de l'art. Le cœur de ces expériences était d'exploiter des embeddings contextuels afin d'améliorer les performances. Trois approches ont été testées afin d'intégrer ces embeddings et de remplacer les embeddings de mots. Les différentes versions atteignent au mieux l'état de l'art. L'article présente quelques pistes déjà étudiées expliquant les raisons pour lesquelles les expériences testées ne dépassent pas le modèle initial et ouvrent des possibilités d'amélioration.

ABSTRACT

Modifying an end-to-end named entity linking model by adding contextual embeddings

This paper presents the experiments performed on a named entity linking system. This task is divided into two main parts, which are the detection of mentions deserving to be linked to the knowledge base and the disambiguation which makes it possible to select the final entity to be linked to each mention. Two approaches exist to solve this task. There are disambiguation-only one and *end-to-end* one that perform both subtasks simultaneously. We were interested in the *end-to-end* model reaching the state of the art results. The aim of these experiments was to experiment the use of contextual embeddings in order to improve performance. Three approaches have been tested to integrate these embeddings and replace word embeddings. The different versions reach the state of the art at best. The article discusses some previously explored avenues of why the experiments tested did not go beyond the initial model and open upgrading possibilities.

MOTS-CLÉS : mention, entité nommée, base de connaissances, approche de bout en bout, vecteurs sémantiques.

KEYWORDS: mention, named entity, knowledge base, end-to-end, embeddings.

Introduction

La tâche de liage d'entités nommées permet d'identifier les éléments d'intérêts dans un texte et de les relier à une entrée d'une base de connaissances. Cette tâche est primordiale pour d'autres applications telles que le résumé de texte, les systèmes de question-réponses ou l'augmentation de bases de connaissances (Shen *et al.*, 2014). C'est cependant une tâche difficile car elle doit à la fois se

charger de repérer les mentions d'intérêts dans un texte et en même temps déterminer à quelle entrée de la base de connaissance chaque mention correspond. Or il est fréquent que relier une mention à une entité soit ambigu. Plusieurs entrées peuvent correspondre et se tromper entraîne alors la génération d'un contre-sens sur le texte analysé. Les deux tâches que sont (a) trouver les mentions puis (b) les relier à une entité peuvent être disjointes, beaucoup de travaux ne concernent que l'un des deux aspects. Néanmoins, effectuer la tâche d'un seul trait, par une approche dite *end-to-end*, peut avoir l'avantage de renforcer les performances globales du modèle. À partir d'un système *end-to-end*, l'objectif de ce travail est d'en modifier l'architecture afin de tenter de l'améliorer. L'axe privilégié a été l'amélioration des embeddings de mots, initialement du Word2Vec (Mikolov *et al.*, 2013), pour les remplacer par des embeddings contextuels de type BERT (Devlin *et al.*, 2018). Le modèle BERT a été choisi car il correspond à l'état de l'art en terme de modèle d'embeddings contextuels. Le travail présenté se base sur un corpus de liage d'entités en anglais.
Cette article présentera dans un premier temps la tâche de liage d'entités nommées ainsi que les travaux récents, puis présentera les différentes expériences réalisées avec BERT avant de présenter les résultats obtenus et de les discuter.

1 Travaux récents

La tâche de **Liage d'Entités Nommées** (LEN) consiste à repérer les mentions d'intérêts dans un document permettant sa compréhension et sa mise en contexte en les reliant à une base de connaissances. Elle est traditionnellement composée de deux sous-tâche (Shen *et al.*, 2014).
La première est **La Reconnaissance d'Entités Nommées** (REN) qui consiste à trouver dans un texte les mots ou groupes de mots significatifs qui doivent être mis en relation avec la base de connaissances, dont les éléments sont appelés *entités*. Ces mots ou groupes de mots sont appelés des *mentions* car ils mentionnent des éléments de la base de connaissances.
La seconde tâche est **La Désambiguïsation d'Entités Nommées** (DEN) qui consiste à relier à chaque mention la bonne entité dans la base de connaissances. C'est une tâche de désambiguïsation car plusieurs entités peuvent initialement correspondre à une même mention. Par exemple, si dans un texte on relève la mention "*le président français*", plusieurs entités issues d'une base de connaissances peuvent correspondre en premier lieu car il y a eu plusieurs présidents français (même si on se limite aux présidents de la République). De même, un texte comportant un nom comme "*Obama*" pourra correspondre à plusieurs entités car plusieurs personnes peuvent porter ce patronyme. Il faut donc déterminer duquel on parle. Cette tâche est souvent découpée en 2 temps (Shen *et al.*, 2014). On commence par générer l'ensemble des entités qui pourraient correspondre à la mention donnée (*Candidate Entity Generation*), puis on les classe de la plus pertinente à la moins pertinente pour prendre la décision d'association finale (*Candidate Entity Ranking*), la pertinence pouvant correspondre à la fréquence (moyen le plus naïf).
Ainsi, un système LEN a besoin d'une **base de connaissances** qui servira de référence pour associer les mentions de tous les textes. *Wikipedia* [1] est souvent retenu pour ce rôle. Chaque page correspond à une entité unique, et il n'existe pas deux pages référençant la même personne, institution, concept ou autre (on exclut les cas des pages traduites). Une bonne base de connaissances n'est pas qu'un simple dictionnaire et possède des liens entre les entités qui permettent de les situer les unes par rapport aux autres. Dans Wikipedia de tels liens peuvent s'exprimer par les *liens hypertextes* présents dans une page et permettant d'atteindre d'autres pages. Ces liens hypertextes étant associés à des mots ou groupes de mot, ils permettent aussi d'avoir des exemples de mentions devant être reliés à ces entités. Ces liens peuvent permettre aussi de définir des relations qui peuvent s'avérer utiles lors

1. http ://www.wikipedia.org/

du classement des entités comme leur fréquence. Les autres bases de connaissances fréquemment utilisées sont YAGO (Fabian *et al.*, 2007), DBPedia (Auer *et al.*, 2007) et Freebase (Bollacker *et al.*, 2008). Le travail sur la base de connaissances est fait en amont, le modèle l'utilise seulement.
Un système LEN a ensuite besoin d'un **corpus de textes** sur lequel le modèle devra extraire et lier les mentions vers les entités de la base de connaissances. Chaque texte du corpus (ou document) est composé de plusieurs phrases et potentiellement de plusieurs mentions afin de les mettre en relation, de les contextualiser et de faciliter ainsi le travail de désambiguïsation. Un corpus de textes peut être un recueil d'articles de presse. Les deux principaux corpus pour le liage d'entités nommées sont CoLNN (Hoffart *et al.*, 2011) et TAC [2] (Getman *et al.*, 2018; McNamee & Dang, 2009).
Certains modèles découplent les parties REN et DEN. La partie DEN est la plus complexe et il existe des systèmes permettant de réaliser la tâches REN en amont (tels que StandfordNER [3] ou OpenNLP [4]). Cependant, depuis quelques années, les modèles tentent davantage une approche *end-to-end* (Shen *et al.*, 2014). L'idée est de rendre la tâche REN plus robuste en utilisant les résultats de la désambiguïsation pour aider à mieux capturer les mentions. Par exemple, un système découplé aura tendance à mal étiqueter *The New York Times* pour le tronquer en *New York Times* ou simplement *New York* (l'article *the* étant le plus souvent oublié). Or, un bon étiquetage de mention peut permettre de faciliter sa correspondance avec une référence de la base de connaissance. DeepType (Raiman & Raiman, 2018) est un exemple de système de *désambiguïsation seule*. Il est la référence de l'état de l'art en DEN. End-to-End Neural Entity Linking (Kolitsas *et al.*, 2018) est un exemple de système *end-to-end*. Il est la référence de l'état de l'art pour cette approche.
Certains systèmes utilisent BERT (Devlin *et al.*, 2018) pour obtenir des *embeddings* contextuels car il peut être spécialisé pour réaliser une tâche de liage d'entités nommées (Broscheit, 2019). D'autres approches se concentrent sur une adaptation de BERT dans le cadre de *datasets* plus spécifiques auxquels BERT n'a pas été confronté, comme le fait PEL-BERT (Li *et al.*, 2020). BERT n'est pas le seul modèle d'*embeddings* contextuels (comme ELMo (Peters *et al.*, 2018) ou GPT (Radford *et al.*, 2019)), mais il constitue la référence.
Nous avons choisi de nous appuyer sur un modèle NEL pré-existant : le *End-to-End Neural Entity Linking* de Kolitsas (Kolitsas *et al.*, 2018). Il effectue la tâche d'*Entity Linking* (EL) de la *détection de mention* (REN) jusqu'à la *désambiguïsation* (DEN). Ce choix a été motivé car le modèle de Kolitsas est à l'état de l'art actuel en tant que système *end-to-end*. Des travaux comme ceux de Broscheit (Broscheit, 2019) l'utilise comme point de comparaison. Son architecture est, de plus, aisée à modifier, contrairement à des modèles comme DeepType qui repose principalement sur son système de types. Ainsi, il était plus envisageable d'explorer l'impact de nouveaux modules en les utilisant sur le modèle de Kolitsas. Enfin, les modules utilisés par ce système sont des outils éprouvés tels que Word2Vec (Mikolov *et al.*, 2013) et des *Bidirectional Long Short-Term Memory* (Bi-LSTM). Il était donc intéressant de tester des améliorations en intégrant des approches plus récentes ayant prouvé leur robustesse (notamment BERT).

2 Modèle Initial

Cette section a pour but d'expliquer le fonctionnement du modèle *End-to-End Neural Entity Linking* de Kolitsas (Kolitsas *et al.*, 2018), illustré dans la Figure 1 afin de comprendre par la suite les modifications qui y ont été apportées. Le modèle part d'un document, soit un texte cohérent, de quelques phrases. Le modèle y détecte les mentions, les relie aux entités correspondantes dans la base de connaissances et renvoie une liste des couples Mention-Entités ainsi prédits dans le document.

2. https ://tac.nist.gov/2010/RTE/
3. https ://nlp.stanford.edu/ner/
4. http ://opennlp.apache.org/

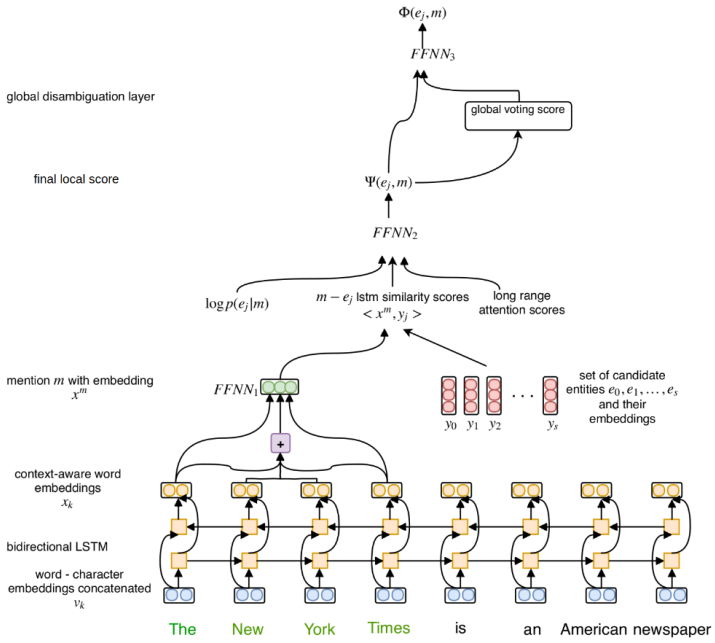


FIGURE 1 – Architecture du modèle End-to-End Neural Entity Linking (Kolitsas *et al.*, 2018)

Le principe général est de construire un **embedding de mention** à partir des embeddings de mots. Cet embedding de mention pourra être comparé à des embeddings d'entités pré-calculés, permettant d'assigner un score de similarité à chacun. Enfin, on choisit la meilleure entité candidate grâce au score de similarité des embeddings et de la cohérence globale en utilisant les autres entités présentes dans le document. L'entrée est constituée des **embeddings de mots *Word2Vec* pré-entraînés**. Les embeddings de mots finaux $\{\nu_k\}_{k\in 1,n}^{-}$ sont obtenus par concaténation des embeddings de mots *Word2Vec* et des embeddings de caractères de chaque mot. Ces embeddings de caractères sont appris grâce à un bi-LSTM appliqué sur les caractères du mot (étape absente sur la figure 1). Ces embeddings de mots sont ensuite transformés en **embeddings de contexte** $\{x_k\}_{k\in 1,n}^{-}$ (appelé *context-aware word embedding* dans la figure 1) grâce à un bi-LSTM. Une fois les embeddings de contexte fixés, ceux pertinents dans la construction d'une mention (c'est-à-dire les embeddings de contexte correspondant à la mention $m = w_q, ..., w_r$) sont combinés grâce à un réseau Feed Foward (appelé ***FFNN de Mention*** ou $FFNN_1$ dans la figure 1) pour transformer l'ensemble en **Embedding de Mention**. Le modèle ne prédéfinit pas les mentions qui seront construites. Il construit et teste toutes les mentions possibles. Compte tenu des notations précédentes, l'Embedding de Mention est obtenu ainsi :

$$x^m = FFNN_1(g^m) \text{où } g^m = [x_q; x_r; \hat{x}^m] \text{ et } \hat{x}^m = \sum_{k=q}^{r} \alpha_k^m \nu_k.$$

Les coefficients α_k^m sont obtenus ainsi :

$$\alpha_k^m = \frac{exp(\alpha)}{\sum_{t=q}^{r} exp(\alpha_k)} \text{ et } \alpha_k = \langle w_\alpha, x_k \rangle$$

Chaque **Embedding de Mention** est ensuite comparé à une liste de candidats $((e_j)_{\geq 1})$ issus des **Embeddings d'Entités** pré-entraînés (Ganea & Hofmann, 2017) $(y_e)_{\in wikipedia}$ et sélectionnés par des probabilités **pré-calculées** (ou *prior* $(p(e_j, m))$ à partir des liens hypertextes de Wikipedia. Ces probabilités ont été établies lors de l'apprentissage des embeddings d'entités et sont considérées comme acquises par le modèle. Chaque couple Entité-Mention reçoit ensuite un score de similarité Ψ obtenu par le réseau Feed Foward (appelé ***FFNN de Score*** ou $FFNN_2$ dans la figure 1) :

$$\Psi(e_j, m) = FFNN_2([logp(e_j, m); \langle x^m; y_i \rangle])$$

Seules les entités avec un score final suffisamment haut sont testées (supérieur à un paramètre γ'). Ceci permet de filtrer les mentions construites par le modèle qui ne correspondent à aucune réelle entité. On obtient donc l'ensemble des couples (mention, candidat) sérieux :

$$V_G = \{(m, e) \in M, e \in (e_j)_{\geq 1}, \Psi(e, m) \geq \gamma'\}$$

Enfin, le modèle procède à une désambiguïsation globale qui permet d'unifier les entités retenues en prenant en compte la cohérence globale entre toutes les entités sélectionnées. On compare donc chaque candidat retenu avec l'ensemble des candidats retenus pour les autres mentions

$$G(e_j, m) = cos(y_{e_j}, y_G^m) \text{ où } y_G^m = \sum_{e \in V_G^m} y_e \text{ et } V_G^m = \{e | (m', e) \in V_G \wedge m' \neq m\}$$

Le vote final consiste à la combinaison par le réseau Feed Foward appelé ***FFNN de Vote***, ou $FFNN_3$ dans la figure 1, de cette comparaison avec le score Ψ :

$$\Phi(e_j, m) = FFNN_3([\Psi(e_j, m), G(e_j, m)])$$

On obtient ainsi en sortie, pour chaque mention initiale du texte, l'entité à laquelle elle fait référence. C'est à partir de cette architecture que nous allons chercher à améliorer le modèle.

3 Expériences et résultats

Cette section présente les travaux réalisés pour remplacer les embeddings de mots Word2Vec (Mikolov *et al.*, 2013) par des embeddings plus robustes. Il a été choisi d'utiliser BERT (Devlin *et al.*, 2018) car il s'agit des embeddings contextuels les plus performants de l'état de l'art. De plus, la première étape du modèle consiste à transformer les embeddings de mots en embeddings de contexte par l'intermédiaire d'un bi-LSTM.

3.1 Protocole expérimental

Trois manières d'utiliser les embeddings contextuels BERT (Devlin *et al.*, 2018) sont explorées.

— La première (*Hypothèse 1*) consiste à remplacer la couche d'embeddings de mots et le bi-LSTM de contextualisation par les embeddings BERT. L'idée est de considérer que les embeddings contextuels de BERT ont déjà l'information initialement produite par le bi-LSTM et peuvent donc le remplacer. Le but est de vérifier s'ils sont plus performants que ceux générés par le modèle.

— La seconde (*Hypothèse 2*) consiste à remplacer les embeddings de mots par les embeddings contextuels BERT. L'idée ici est de considérer que les embeddings contextuels BERT peuvent être assimilés à des embeddings de mots (et non plus contextels) plus puissants que ceux de Word2Vec. Ils passent donc par le bi-LSTM afin de créer du contexte.

— La dernière (*Hypothèse 3*) est de considérer que les embeddings BERT peuvent apporter de l'information supplémentaire et venir en soutien du bi-LSTM. L'idée est de combiner les embeddings BERT aux embeddings de contexte issus du bi-LSTM en supposant que la connaissance issue de BERT va ainsi améliorer la qualité de l'embedding contextuel global.

Les trois méthodes ont donc été testées selon les protocoles décrits ci-dessous. Les expériences ont toutes été menées sur 50 itérations. Les autres paramètres d'apprentissage ont été conservés identiques à ceux originellement utilisés par Kolitsas (Kolitsas *et al.*, 2018) à l'exception des modifications précisées.

3.1.1 Utilisation des embeddings en tant qu'embeddings de mots

Comme illustré dans la figure 2, le modèle subit peu de modifications pour l'*Hypothèse 1*. On remplace uniquement les embeddings de mots initiaux (Word2Vec) par ceux extraits depuis BERT.

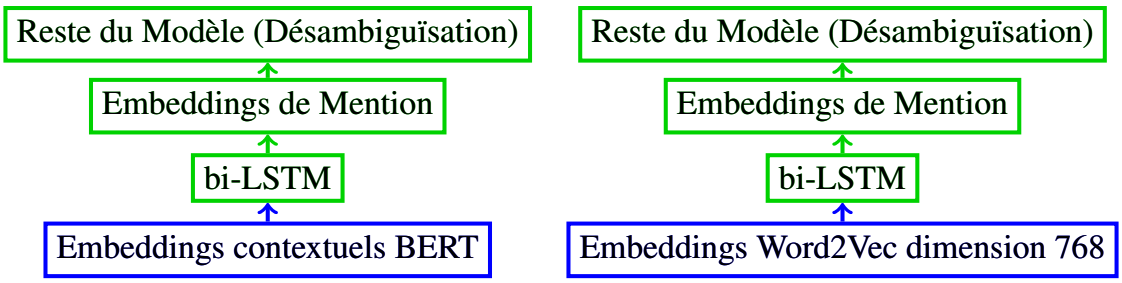


FIGURE 2 – `WordBERT`, à gauche, et équivalent `Word2Vec768` à droite

Seuls les embeddings de caractères ont été retirés du processus (WordBERT de gauche).
Afin de comparer réellement l'impact des embeddings BERT sur le résultat et le séparer de l'impact des changements d'architecture (c'est-à-dire la taille des embeddings et la suppression des embeddings de caractère), une version identique avec Word2Vec a été également testée (Word2Vec768). De plus, pour des raisons techniques, certains documents ont été retirés, car il était impossible de générer des embeddings BERT pour ces derniers. La figure 3 donne un exemple de tel document. Cela correspond à 1485 entités (8%) sur l'ensemble d'entraînement, 298 (6%) sur AIDA Test A et 314 (7%) sur AIDA Test B. Afin d'éviter des biais, le modèle initial (Kolitsas *et al.*, 2018) a été ré-entraîné en enlevant ces exemples supprimés.

```
TENNIS - FRIDAY 'S RESULTS FROM THE U.S. OPEN . NEW YORK 1996-08-30 Results from the U.S. Open Tennis Championships at the National Tennis Centre on
Friday ( prefix number denotes seeding ) : Women 's singles , third round Sandrine Testud ( France ) beat Ines Gorrochategui ( Argentina ) 4-6 6-2 6-1
Men 's singles , second round 4 - Goran Ivanisevic ( Croatia ) beat Scott Draper ( Australia ) 6-7 ( 1-7 ) 6-3 6-4 6-4 Tim Henman ( Britain ) beat Doug
```

FIGURE 3 – Exemple de document retiré du corpus AIDA non tokenisable par BERT

3.1.2 Utilisation des embeddings BERT en tant qu'embeddings contextuels

Comme illustré dans la figure 4, le modèle subit une modification plus importante pour l'*Hypothèse 2*. On supprime la couche de bi-LSTM pour connecter directement la couche de génération de l'embedding de mention avec les embeddings contextuels BERT. On retire les embeddings de caractères, non pertinents si l'on considère que BERT possède déjà l'information nécessaire. On supprime enfin le mécanisme d'attention qui liait les embeddings d'entités aux embeddings de contexte (en sortie du bi-LSTM). Le reste du modèle est inchangé.

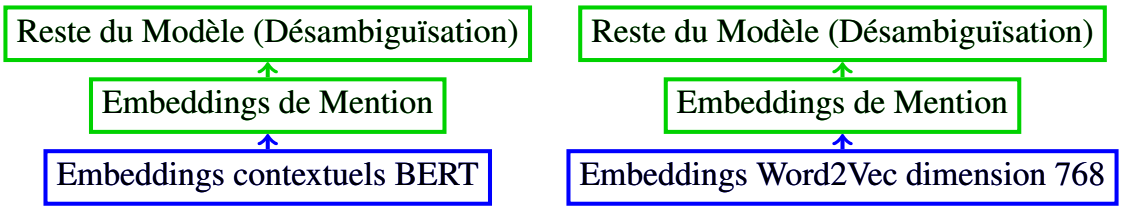


FIGURE 4 – `ContextBERT` à gauche, et équivalent `Context Word2Vec` à droite

3.1.3 Association des embeddings de BERT avec les embeddings contextuels du modèle

Comme présenté dans la figure 5, l'objectif dans l'*Hypothèse 3* est d'ajouter des embeddings contextuels BERT à ceux de contexte issus du bi-LSTM. Pour cela, l'architecture initiale est conservée, mais en concaténant les embeddings contextuels BERT aux embeddings de contexte du bi-LSTM correspondants.

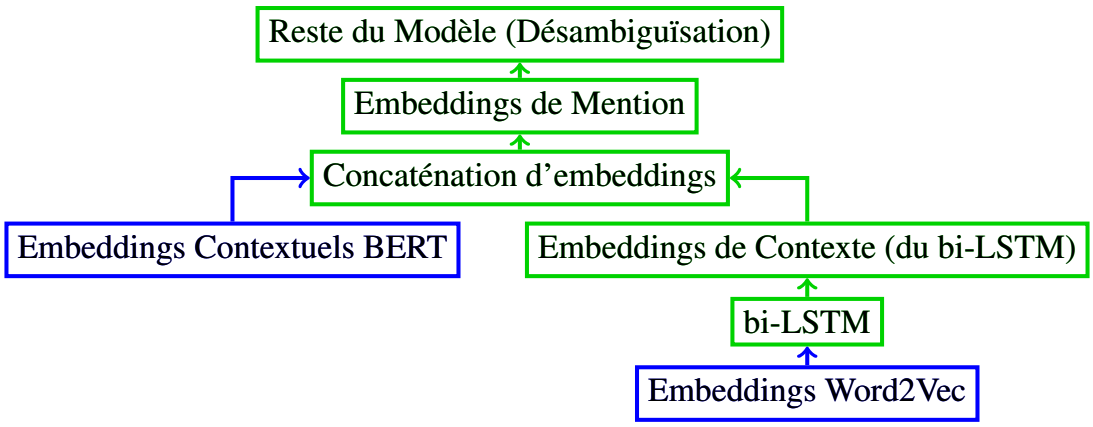


FIGURE 5 – Expérience BERT+LSTM

3.2 Données utilisées

Ces différentes expériences ont été menées sur le même corpus que celui utilisé par Kolitsas. Il s'agit du corpus AIDA/CoNLL (Hoffart *et al.*, 2011) qui est le plus gros corpus public de liage d'entités nommées en anglais. Il est composé d'un ensemble d'entraînement de 18448 mentions liées dans 946 documents. L'ensemble de validation (AIDA test A) contient 4791 mentions dans 216 documents et l'ensemble de test (AIDA test B) contient 4485 mentions dans 231 documents.
Les variantes BERT ont été obtenues en extrayant les embeddings depuis la version pré-entraînée de BERT de base pour chaque document du corpus AIDA/CoNLL. Comme BERT renvoie 12 couches d'embeddings, il faut choisir lesquelles considérer comme embeddings contextuels. Plusieurs méthodes existent comme indiqué dans (Devlin *et al.*, 2018).
Après expérience, nous avons choisi de sommer les quatre dernières couches ce qui donne de meilleures performances par rapport aux autres méthodes de combinaison des couches BERT (Devlin *et al.*, 2018).

3.3 Résultats et Analyses

Approche	mic/mac F1 AIDA Test A	mic/mac F1 AIDA Test B
Modèle de Base (Kolitsas *et al.*, 2018)	86.6 / 89.4	82.6 / 82.4
BERT fine-tuné (Broscheit, 2019)	87.3 / 92.3	79.3 / 81.1
Modèle Réajusté	89.7 / 87.6	**85.5** / 84.8
Word BERT	87.1 / 84.3	83.7 / 83.2
Context BERT	85.9 / 81.8	83.3 / 81.7
BERT + LSTM	89.2 / 87.1	85.4 / **84.9**
Word2Vec 768	83.0 / 79.8	79.9 / 76.4
Context Word2Vec	68.9 / 66.1	62.6 / 61.2

TABLE 1 – micro / macro F1 des différentes utilisations de BERT (en **gras** les meilleurs résultats et en vert les seconds meilleurs)

Les modèles sont évalués en termes de macro et micro F1. La micro F1 est la F1 calculée sur l'ensemble des documents du dataset. La macro F1 est la moyenne des F1 sur les différents documents du dataset. Un modèle stable dans ses prédictions sur les différents documents présentés aura donc une micro et une macro F1 similaires.

Une analyse de la signification des performances a été menée afin de déterminer à partir de quel seuil on peut dire qu'il y a une amélioration de résultat, certains biais empêchant une reproductibilité exacte des résultats. Chaque expérience a donc été lancée dix fois pour calculer un écart-type entre les différents résultats de chaque expérience puis un écart-type des différences entre toutes les expériences. Chaque itération de chaque expérience étant identique aux autres, les écarts de résultats traduisaient une variabilité qui ne pouvait pas être interprétée comme étant une amélioration du modèle. Cette analyse a montré que deux expériences ayant des performances avec un écart inférieur ou égal à 0.3 points de F1 pouvaient être considérées comme étant équivalentes.
Les résultats des expériences sont présentés dans le tableau 1. Les expériences reportées sont représentatives de celles répétées plusieurs fois sans en être avoir des résultats marginaux. Les analyses des résultats s'appuient exclusivement sur les expériences reportées dans le tableau 1. Le Modèle de Base correspond aux résultats du modèle de Kolitsas (Kolitsas *et al.*, 2018) tel qu'il est proposé tandis que le Modèle Réajusté correspond au même modèle entraîné sur le dataset tronqué des exemples non utilisés dans les autres expériences. C'est donc à ce modèle que l'on compare les résultats. BERT fine-tuné correspond à l'expérience de Broscheit consistant à utiliser directement BERT comme système de liage d'entités nommées. Il se compare lui-même à Kolitsas dans son article. Word BERT utilise un vocabulaire tel que chaque occurrence de chaque mot soit présente de manière distincte. Word2Vec 768 conserve l'architecture initiale adaptée pour les embeddings BERT mais en utilisant des embeddings Word2Vec de dimensions similaires à BERT (768). Context BERT utilise un vocabulaire tel que chaque occurrence de chaque mot soit présente de manière distincte. Les embeddings BERT sont directement utilisés pour la génération de l'embedding de mention. Context Word2Vec conserve l'architecture initiale adaptée pour les embeddings BERT mais en utilisant des embeddings Word2Vec de dimensions similaires à BERT (768). Le BERT + LSTM correspond au modèle où l'on vient concaténer les embeddings BERT aux embeddings de contexte du bi-LSTM en sortie de ce dernier. Le reste du modèle est inchangé.

3.3.1 Analyse Quantitative

Dans le tableau 1, on remarque que le Modèle Réajusté (*85.5 / 84.8*) est équivalent à BERT + LSTM (*85.4 / 84.9*). On observe également que Word BERT (*83.7 / 83.2*) est légèrement supérieur à Context BERT (*83.3 / 81.7*). On peut en conclure que les embeddings BERT ne possèdent initialement pas plus d'information que ce que peut fournir le bi-LSTM au cours de son apprentissage. En revanche, ils semblent plus adéquats en étant utilisés comme embeddings de contexte secondaires, pour appuyer les résultats du bi-LSTM.
Ensuite, on constate une forte différence entre le Modèle de Base (*82.1 / 83.6*) et le Modèle Réajusté (*85.5 / 84.8*). Cela signifie que les éléments retirés du datasets sont des éléments sur lesquels le modèle initial de Kolitstas se trompait. En les oberservant on remarque qu'il s'agit en effet pour la plupart de tableaux de résultats sportifs (exemple figure 3) qui ne correspondent pas à un texte narratif où l'on peut se reposer sur le contexte entre les mots pour inférer son sens.
On constate alors que Word BERT (*83.7 / 83.2*) ne permet pas d'atteindre les performances initiales (*85.5 / 84.8*). En revanche, la même expérience testée avec les embeddings Word2Vec (c'est-à-dire Wor2Vec 768) obtient des résultats encore plus faibles (*79.9 / 76.4*).
Context Word2Vec est très significativement moins bon (*62.6 / 61.2*). C'est un résultat attendu car on y considère des embeddings de mots comme embeddings de contexte. Cela a du sens de partir du principe que les embeddings BERT possèdent une information contextuelle car ils ont été appris dans ce but.
En revanche, il n'y a pas de raison que des embeddings de mots possède une information contextuelle. L'expérience permet néanmoins de mettre en évidence l'importance du bi-LSTM dans le traitement des embeddings de mots.

De manière générale, aucune des expériences n'atteint ou ne dépasse significativement les performances initiales du modèle. Plusieurs pistes d'explications ont été explorées.

3.3.2 Analyses Qualitatives

Nous avons commencé par regarder plus en détail les prédictions faites par le modèle et les différentes variantes BERT. Les analyses se sont focalisés sur WordBERT, BERT+LSTM et le Modèle de Base Réajusté. Afin de les mener, les mots proches des entités mal prédites ont été visualisées via une représentation T-SNE (exemple figure 6). Toutes les visualisations (figure 6 à figure 11) sont faites pour dix entités mal prédites par les trois variantes du modèle testé, sur le dataset AIDA Test A, ayant soit le plus haut, soit le plus bas score de similarité. La légende indique, pour chaque entité représentée, le mot issu de la mention choisi pour représenter l'entité, l'entité prédite par le modèle et enfin l'entité de référence qui aurait dû être prédite. Concrètement, pour chaque entité mal prédite, le premier mot de la mention menant à cette prédiction a été récupéré. Puis, les entités ont été triées en fonction du score de similarité attribué par le modèle entre la mention et l'entité. Dix entités ont été sélectionnées, soit avec le plus haut score de similarité, soit le plus faible. Les dix mots les plus proches sémantiquement du mot menant à l'entité prédite ont enfin été affichés.
Cette procédure a été réalisée en extrayant indépendamment pour chacun des 3 modèles les embeddings de mots puis dans un second temps les embeddings de contexte, c'est-à-dire à la sortie du bi-LSTM ou – pour le cas de l'expérience BERT+LSTM – juste avant la génération de l'embedding de mention. Ces analyses ont permis d'avancer certaines hypothèses.
Les figures 6 à 9 ne présentent pas les résultats pour le Modèle Réajusté mais sont équivalents. La figure 6 montre les embeddings de mots qui sont identiques entre le Modèle Réajusté et BERT+LSTM. De même, que la nature des embeddings de mots ne changent pas le traitement du bi-LSTM, ce qui s'est retrouvé dans les figures du Modèle Réajusté et de WordBERT.
Les figures 6 à 11 sont données en annexe.

Répartition des entités et impact du bi-LSTM

Les entités mal prédites ont été récupérées afin de les comparer en fonction de l'ensemble dont elle provenait. Les entités erronées sont réparties de manière homogène dans les datasets. C'est-à-dire qu'en prenant en compte le nombre d'occurrences de chaque entité mal prédite, il n'y a pas d'entité qui prend une place significative parmi celles-ci. Le maximum est à 2.5% de l'ensemble des occurrences apparaissant dans les erreurs et le minimum à 0.1% quand il n'y a qu'une seule occurrence de l'entité. Ainsi, la moitié des erreurs sur AIDA Test A et les $2/3$ des erreurs sur AIDA Test B sont sur des entités qui ne se trouvaient pas dans l'ensemble d'entraînement.
En observant les figures générées à partir des embeddings de mots (figure 6 et 8) et celles à partir des embeddings de contexte (figure 7 et 9), on peut visualiser l'impact du bi-LSTM sur l'adaptation des embeddings de mots afin de les faire coïncider avec le contexte du document. Certaines des erreurs de prédictions d'entités sont liées à une mauvaise désambiguïsation de la mention. Par exemple, on voit l'évolution du contexte entourant le mot `China` dans les figures 8 (embeddings de mots) et 7 (embeddings de contexte). Ce terme doit mener à la prédiction de l'entité `Qing Dynasty`. Dans la figure 8, ce dernier est très proche du terme `China`. On a peu d'information sur le contexte dans lequel il se trouve. Or, le bi-LSTM va créer du contexte et on observe dans la figure 7 que les termes proches sont désormais des noms de pays tels que `Spain`, `Norway` ou `Nigeria`. On perd ainsi tout contexte relatif à une dynastie impériale au profit seul d'un pays, ce qui correspond à la prédiction du modèle. Ces erreurs sont principalement des exemples qui se trouvent dans le dataset AIDA Test A mais pas dans le dataset d'entraînement.
En observant de plus les figures 6 à 9 sur les variantes BERT, on peut remarquer que le bi-LSTM ne produit aucun contexte sur la variante BERT + LSTM. En effet, on peut observer ceci en comparant

les figures sur la variante WordBERT (figure 6 et 7) et BERT + LSTM (figure 8 et 9). Dans la figure 6, on a la représentation d'embeddings BERT seuls qui prennent ensuite un contexte produit par le bi-LSTM dans la figure 7. Or, cette représentation est très similaire à celle des embeddings de contexte de BERT + LSTM (figure 9), et plus similaire qu'avec la représentation typique après le bi-LSTM. Cela montre que les informations apprises par le bi-LSTM sont diluées par l'ajout des embeddings BERT qui sert de contexte seul. Cependant, l'usage d'embeddings BERT comme seul contexte ne fonctionne pas entièrement comme l'a montré la différence de performance entre BERT + LSTM (*85.4 / 84.9*) et ContextBERT (*83.3 / 81.7*). Le mécanisme d'attention porté sur les embeddings de mots, présent dans la version BERT + LSTM mais absent dans la version ContextBERT, pourrait dans ce cas expliquer cette différence.

Erreurs liées au seuil de validation

Jusque là, les observations se sont portées que sur les prédictions erronées par les trois principales variantes. Or, en prenant les mauvaises prédictions effectuées uniquement par un modèle, on remarque que beaucoup d'entités sont en réalité correctement prédites (exemple figure 10). Cet effet s'accentue lorsque l'on regarde les entités mal prédites avec un score de similarité faible (exemple figure 11) où la quasi totalité des entités sont correctement prédites. Le modèle s'évaluant simultanément sur la tâche de repérage des mentions et de désambiguïsation, cela aurait pu souligner des erreurs de positionnement des mentions. Or, l'évaluation autorise une position des mentions permissive de sorte que seule une mention entièrement extérieure à celle attendue sera décomptée comme fausse. De plus, les figures 10 et 11 montre que le début des empans est cohérent avec l'entité prédite, ce qui exclut l'hypothèse d'une mention mal positionnée. L'entité est donc bien correctement placée et prédite. Le seul élément pouvant la discréditer reste le seuil de validation d'une prédiction. Ce seuil permet d'exclure toutes les prédictions qui se feraient sur des éléments qui n'ont pas à être prédits et est appris par le modèle en fonction des documents qu'ils voient durant son apprentissage. Si une entité est correctement prédite avec un score en dessous de ce seuil, elle sera décomptée comme une errance du modèle. Ces cas, présents également sur WordBERT et BERT+LSTM, représentent 40% des erreurs de prédiction sur le dataset d'entraînement et 45% sur les datasets AIDA Test. On remarque donc que les erreurs de prédiction se répartissent entre les erreurs communes surtout représentées par des entités mal désambiguïsées, et les erreurs spécifiques à chaque modèle surtout représentées par des scores de similarité trop faibles.

Conclusion

Dans cet article, nous avons présenté la tâche de liage d'entités nommées et nous nous sommes intéressés plus en détail au modèle *End-to-End Neural Entity Linking* (Kolitsas *et al.*, 2018). Ce modèle permet de réaliser la tâche selon une approche end-to-end de la détection de mention jusqu'à la désambiguïsation. Il est au niveau de l'état de l'art et nous nous sommes donc intéressés à l'étudier plus en profondeur pour voir comment l'améliorer. Cet objectif était motivé par l'architecture basique du modèle, n'utilisant que des outils éprouvés. Il y avait donc un potentiel d'amélioration en usant d'approches plus récentes. La piste explorée ici fut l'intégration d'embeddings contextuels BERT (Devlin *et al.*, 2018) pour remplacer les embeddings de mots Word2Vec (Mikolov *et al.*, 2013) initialement présents. Malheureusement, les différents modèles testés n'ont pas apporté d'amélioration significative. Plusieurs pistes ont été envisagées et seront suivies pour expliquer la sous-performance des embeddings BERT, parmi lesquelles réduire l'impact de la taille des embeddings ou l'architecture globale. Cela pourra être dans l'optique de réussir à obtenir les mêmes performance entre le modèle initial et Word768 pour appliquer les modification aux versions avec BERT. Dans la suite de ce premier travail, nous allons analyser la répartition sémantique des mots afin de comprendre d'où proviennent

les erreurs de prédiction d'entités, aussi bien du côté initial que du côté des versions utilisant BERT. Enfin, nous modifierons l'architecture du modèle. Le point clé étant la transformation des embeddings de mots en embeddings de contexte, l'approche envisagée sera de changer le bi-LSTM se chargeant de cette tâche par un transformer.

Références

AUER S., BIZER C., KOBILAROV G., LEHMANN J., CYGANIAK R. & IVES Z. (2007). Dbpedia : A nucleus for a web of open data. In *The semantic web*, p. 722–735. Springer.

BOLLACKER K., EVANS C., PARITOSH P., STURGE T. & TAYLOR J. (2008). Freebase : a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, p. 1247–1250.

BROSCHEIT S. (2019). Investigating entity knowledge in bert with simple neural end-to-end entity linking. *In Proceedings of the 23rd Conference on Computational Natural Language Learning, 677–85. Association for Computational Linguistics, 2019.*

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

FABIAN M., GJERGJI K., GERHARD W. *et al.* (2007). Yago : A core of semantic knowledge unifying wordnet and wikipedia. In *16th International World Wide Web Conference, WWW*, p. 697–706.

GANEA O.-E. & HOFMANN T. (2017). Deep joint entity disambiguation with local neural attention. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2619–29. Association for Computational Linguistics, 2017.*

GETMAN J., ELLIS J., STRASSEL S., SONG Z. & TRACEY J. (2018). Laying the groundwork for knowledge base population : Nine years of linguistic resources for tac kbp. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

HOFFART J., YOSEF M. A., BORDINO I., FÜRSTENAU H., PINKAL M., SPANIOL M., TANEVA B., THATER S. & WEIKUM G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 782–792.

KOLITSAS N., GANEA O.-E. & HOFMANN T. (2018). End-to-end neural entity linking. *In Proceedings of the 22nd Conference on Computational Natural Language Learning, 519–29. Association for Computational Linguistics, 2018.*

LI S., CUI W., LIU Y., MING X., HU J., HU Y. & WANG Q. (2020). Pel-bert : A joint model for protocol entity linking. *ArXiv*, **abs/2002.00744**.

MCNAMEE P. & DANG H. T. (2009). Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, p. 111–113.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv :1310.4546*.

PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. *arXiv preprint arXiv :1802.05365*.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, **1**(8), 9.

RAIMAN J. R. & RAIMAN O. M. (2018). Deeptype : multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

SHEN W., WANG J. & HAN J. (2014). Entity linking with a knowledge base : Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, **27**(2), 443–460.

Annexe

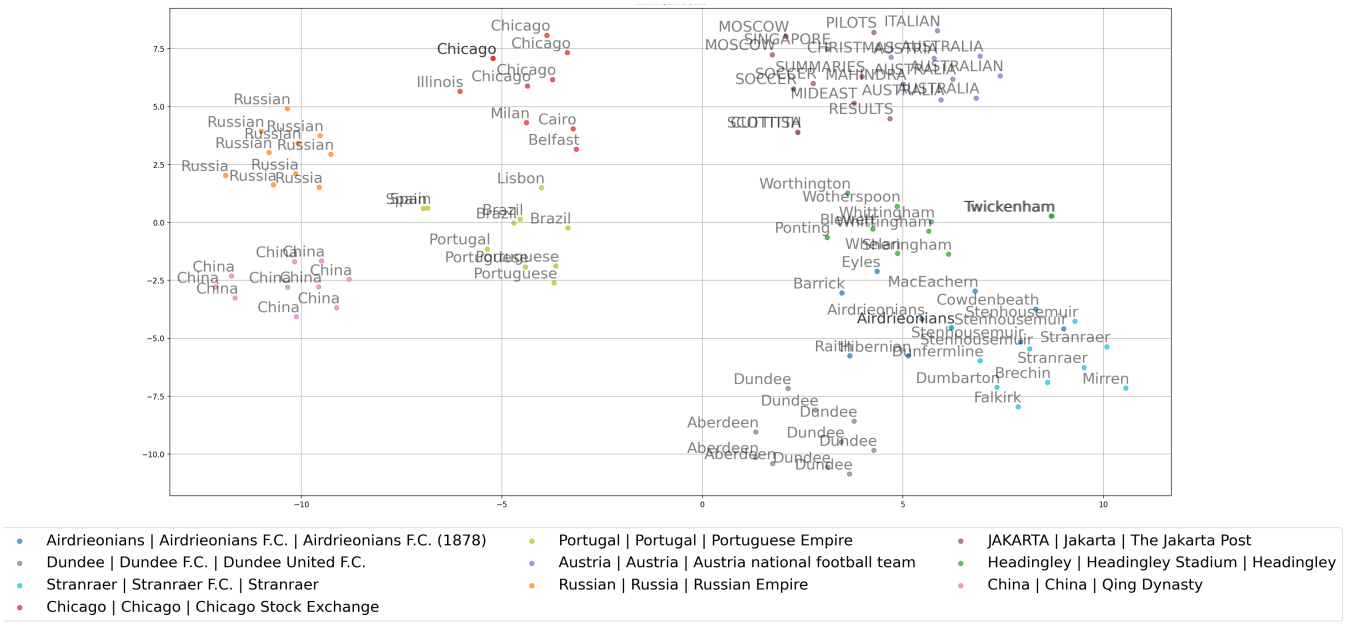


FIGURE 6 – Embeddings de mots de WordBERT, entités mal prédites, plus haut score de similarité.

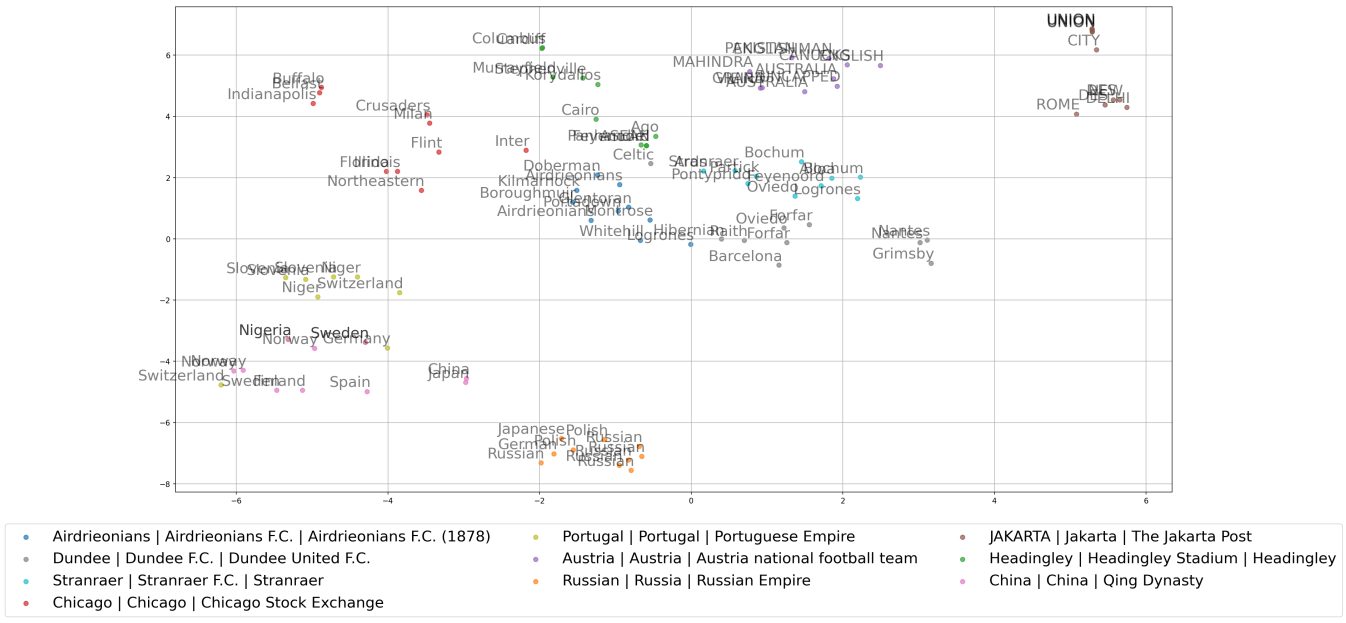


FIGURE 7 – Embeddings de contexte WordBERT, entités mal prédites, plus haut score de similarité.

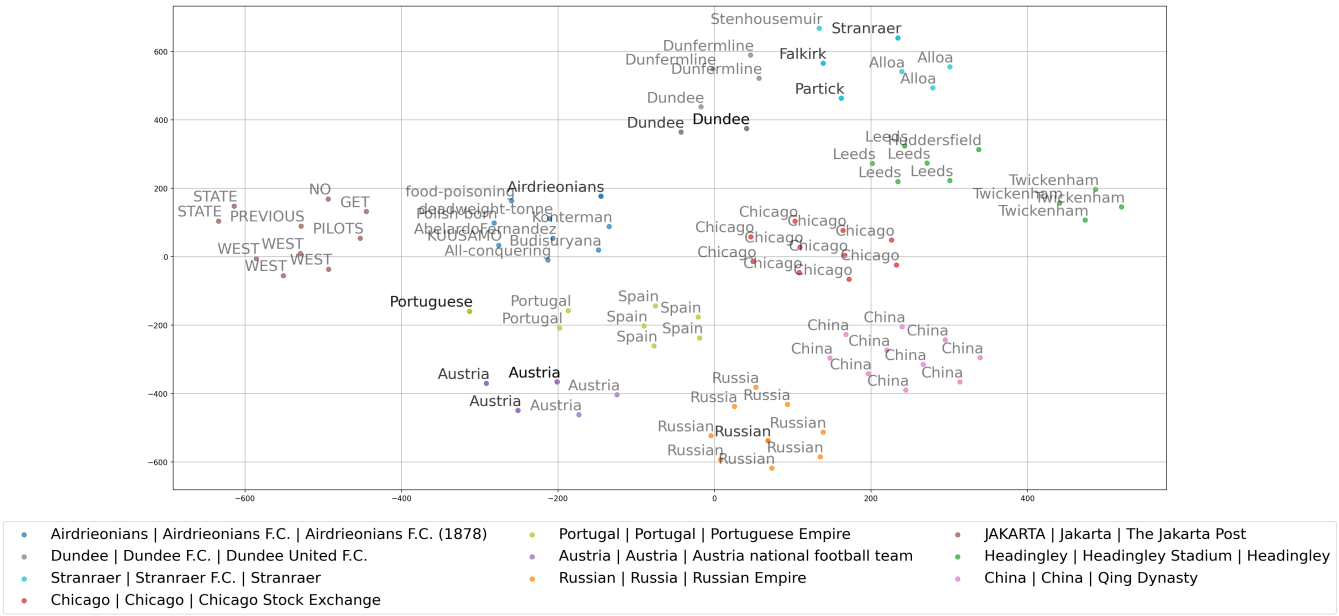


FIGURE 8 – Embeddings de mots BERT+LSTM, entités mal prédites, plus haut score de similarité

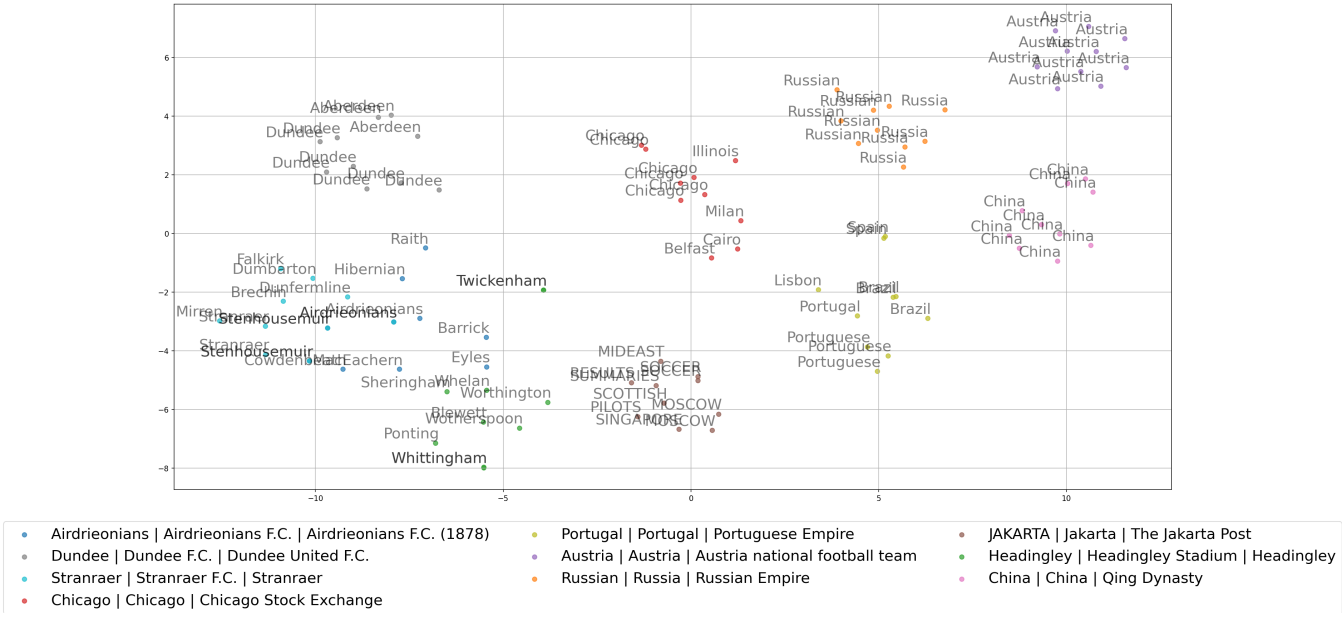


FIGURE 9 – Embeddings de contexte BERT+LSTM, entités mal prédites, plus haut score de similarité.

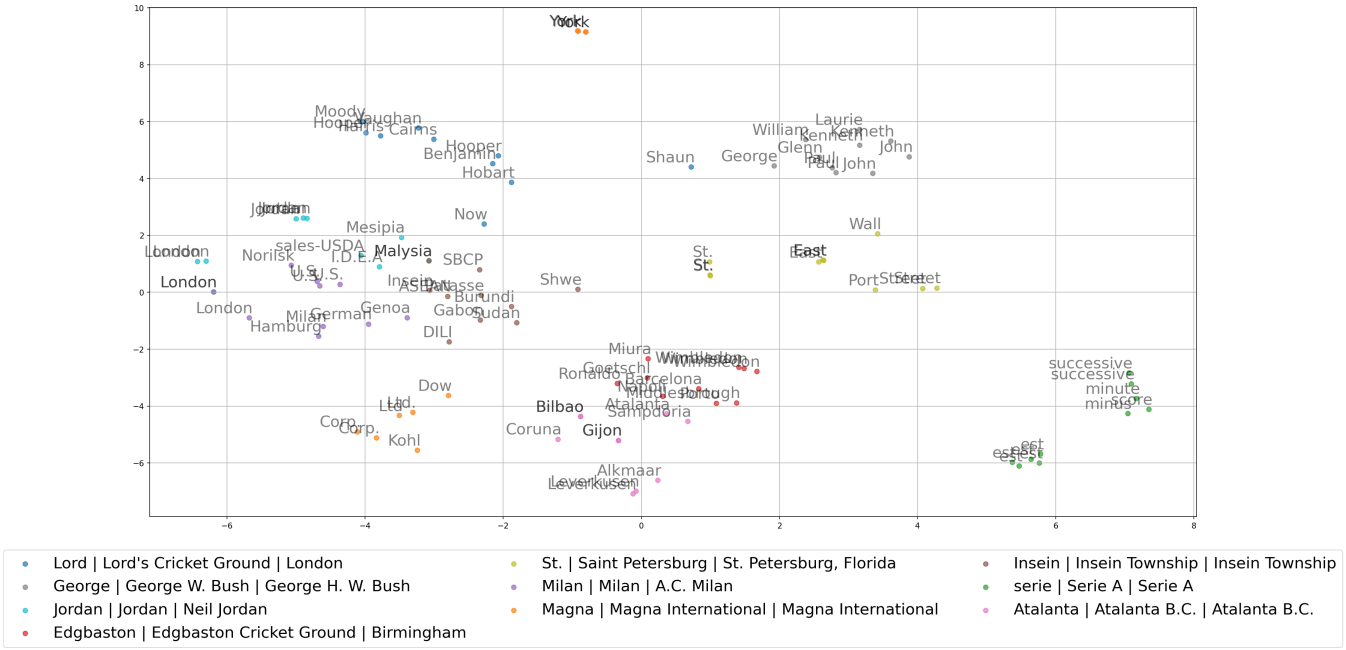


FIGURE 10 – Embeddings de contexte du Modèle Réajusté, entités mal prédites, plus haut score de similarité.

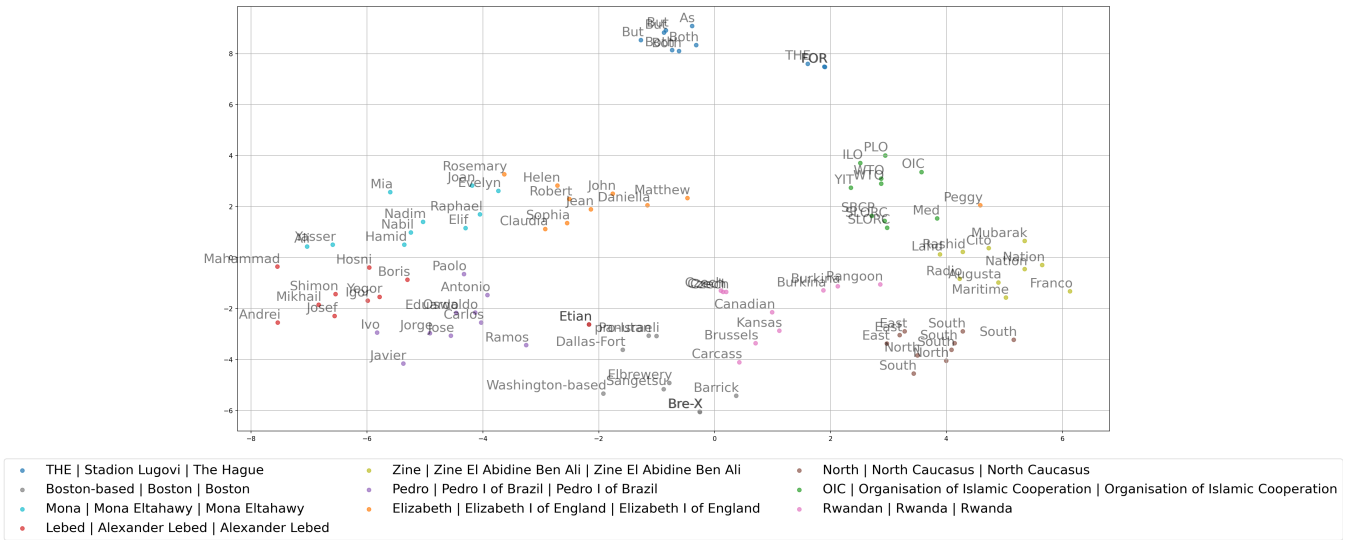


FIGURE 11 – Embeddings de contexte du Modèle Réajusté, entités mal prédites, plus bas score de similarité.

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

Revue de la littérature : entrepôts de données biomédicales et traitement automatique de la langue

Adrien Bazoge[1, 2]
(1) LS2N, UMR CNRS 6004, Université de Nantes, France
(2) CHU de Nantes, INSERM CIC 1413, Pôle Hospitalo-Universitaire 11 : Santé Publique, Clinique des données, Nantes, France
adrien.bazoge@univ-nantes.fr

RÉSUMÉ

La quantité de données de santé informatisées ne cesse de croître et ouvre de nouvelles possibilités pour la recherche scientifique. L'accès à ces données passe très souvent par l'utilisation d'entrepôts de données biomédicales, déployés pour cet usage. Parmi les données stockées dans ces entrepôts, on peut trouver des données textuelles, en plus ou moins grande quantité. Le traitement automatique de la langue (TAL) est le domaine de prédilection pour l'exploitation des données textuelles. Cet article propose une revue de la littérature qui s'intéresse, à travers les publications sur PubMed, ACL Anthology et Google Scholar, à l'interaction entre deux thématiques : les entrepôts de données biomédicales et le traitement automatique des langues. Cette revue montre que l'intérêt pour les données de santé et les entrepôts de données biomédicales est en constante croissance dans la littérature. Elle montre également que le TAL devient peu à peu un outil indispensable afin d'exploiter au mieux les entrepôts de données biomédicales.

ABSTRACT

Literature review : biomedical data warehouse and natural language processing

The amount of electronic health data continues to grow and its availability for research purposes opens up new era of secondary uses of biomedical data. In care organisations, access to these data is mediated by biomedical data warehouses. Biomedical Data warehouses have been recently deployed and recognized as an value creation instrument in care organisations through data. Among the large diversity of data, textual information can be found, in great quantities. To facilitate handling of textual information, natural language processing (NLP) algorithm are increasingly used in biomedical data warehouse. This review present a systematic literature analysis of publications from sources (PubMed, ACL Anthology and Google Scholar) in the interaction between three themes : computerization of health data, biomedical data warehouses and natural language processing. This review shows that the interest in health data and biomedical data warehouses is exponentially growing in the literature. It also shows that NLP is a pivotal tool of data access, extraction and transformation in biomedical data warehouses in all fields of modern medicine.

MOTS-CLÉS : revue de la littérature, entrepôt de données biomédicales, traitement automatique de la langue.

KEYWORDS: literature review, biomedical data warehouse, natural language processing.

Volume 2 : 23e REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL), pages 96–109.

1 Introduction

Depuis 20 ans, les données de santé issues du soin des patients sont systématiquement archivées. Les bases de données ainsi constituées, souvent électroniquement, rassemblent à la fois des données structurées (biologie, démographies, etc.) et des données non structurées (comptes rendus textuels d'hospitalisation ou de consultation). L'intentionnalité première de ces données est l'acte de soin au sens large, leur usage est celui du soin et non de la recherche biomédicale. La recherche biomédicale est un secteur où la source traditionnelle de données chez l'homme est un essai clinique ou un registre de pathologie. Cette masse de données est au carrefour de multiples contributions : celle du patient, pour lequel les données sont collectées lors de l'hospitalisation ou de la consultation ; celles des soignants, qui s'occupent des patients et permettent la collection de ces données ; et celles de l'établissement de santé, qui organise toute la logistique opérationnelle et financière autour du soin et de ses données. Dans un premier temps utilisées pour le soin, ces données peuvent désormais être exploitées à des fins secondaires, pour la recherche et l'évaluation des soins, postérieurement à leur enregistrement, grâce aux avancées technologiques en matière d'intelligence artificielle (IA) appliquées sur des grandes masses de données (*big data*). Cette grande quantité de données qui devient accessible, et notamment les données textuelles, renforce l'intérêt de l'application du traitement automatique des langues (TAL) qui met en oeuvre des algorithmes permettant d'opérer à une échelle aussi massive que les données elles-mêmes (Daille & Nazarenko, 2017).

Dans cette revue de la littérature, nous étudions l'évolution de l'application du TAL dans les entrepôts de données biomédicales depuis l'informatisation des données de santé, à travers les publications sur des moteurs de recherche de la littérature scientifique : PubMed [1], ACL Anthology [2] et Google Scholar [3]. Bien que l'application du TAL soit répandue sur les dossiers patient informatisés de manière générale, nous nous intéressons ici uniquement à l'application du TAL sur des données issues des entrepôts de données de santé. En effet, la recherche sur données de santé est de plus en plus réglementée afin de mieux préserver la confidentialité des patients à l'origine de ces données. Pour permettre cette réglementation, l'exploitation de ces données pour la recherche passe désormais davantage par les entrepôts de données de santé, conçus pour cet usage. L'objectif de cette revue de la littérature est donc d'analyser l'évolution de l'application du TAL sur les données de santé issues des entrepôts de données biomédicales. Cet article est structuré en trois sections. La première section définit les thématiques. Notre méthodologie est présentée dans la deuxième section et aborde la construction des requêtes sur les moteurs de recherche et la définition des axes de classification des publications. Ensuite, la dernière section rassemble les résultats des requêtes et leur analyse.

2 Entrepôts de données biomédicales

Cette revue de littérature est restreinte au champ des entrepôts de données biomédicales pour laquelle nous appliquons la définition suivante :

Un entrepôt de données de santé (*Health Data Warehouse*), aussi appelé entrepôt de données biomédicales ou entrepôt de données cliniques (*Biomedical Data Warehouse*, *Clinical Date Warehouse*), est une base de données relationnelle regroupant une partie ou l'ensemble des données d'une base

1. https://pubmed.ncbi.nlm.nih.gov/
2. https://www.aclweb.org/anthology/search/
3. https://scholar.google.com/

de données opérationnelle dans un établissement de soin. Les entrepôts de données peuvent être construits à partir de plusieurs sources de données *via* un processus dit ETL (*extract, transform, load*). Les entrepôts de données sont ensuite utilisés pour le pilotage de l'activité ou son évaluation à travers les statistiques et l'analyse de données. L'explosion de la production de données numériques a été le facteur permettant de démocratiser la construction et l'utilisation des entrepôts de données.

Le domaine de la santé a également tardé à intégrer en profondeur cette transition numérique. Bien que les entrepôts de données soient installés dans le paysage clinique anglo-saxon depuis plus de dix ans, ce n'est qu'après l'obtention de l'autorisation de la CNIL [4] que les premiers entrepôts de données biomédicales voient le jour en France pour une utilisation à des fins de recherche. L'AP-HP [5] est le premier établissement à obtenir cette autorisation en janvier 2017, suivie par le CHU de Nantes en juillet 2018 et le CHU de Lille en septembre 2019. Les entrepôts de données biomédicales rassemblent les données de millions de patients traités dans les établissements hospitaliers. Les données contenues dans ces entrepôts sont de natures diverses : des données démographiques, des données du PMSI [6], des résultats de biologie et d'imagerie, des prescriptions de médicament ou encore des comptes rendus médicaux de consultation ou d'hospitalisation. À titre d'exemple, l'Entrepôt de Données Biomédicales Nantais (EDBN) rassemble les données de 2,7 millions de patients pour 30 millions de documents.

Dans le cadre de l'exploitation des entrepôts de données biomédicales, ces données sont utilisées à des fins de recherche et peuvent permettre d'améliorer l'efficacité des systèmes de santé, la vigilance et la veille sanitaire.

3 Thèmes de l'analyse

À l'aide des moteurs de recherche de la littérature scientifique PubMed, ACL Anthology et Google Scholar, nous nous intéressons aux travaux publiés entre 1995 et 2020, à travers trois thématiques : (i) l'informatisation des données de santé, (ii) les entrepôts de données de santé et (iii) le traitement automatique de la langue. La thématique d'« informatisation des données de santé » fait référence à la transition numérique, la constitution de bases de données pour stocker les données de santé des patients (Moore *et al.*, 2021). Pour chacune de ces thématiques, une liste de mots clés a été établie :

1. Informatisation des données de santé : *electronic medical record, EMR, electronic health record, EHR, real world evidence, real world data*

   Les mots clés « *electronic medical record* » et « *electronic health record* » font références aux dossiers patient informatisés qui peuvent être exploités dans les études, tandis que les mots clés « *real world data* » et « *real world evidence* » correspondent plutôt aux données de soin des patients qui sont générées au cours de la pratique clinique de routine.

2. Entrepôts de données de santé : *clinical data warehouse, biomedical data warehouse, health data warehouse*

   Les mots clés sélectionnés pour représenter la thématique « Entrepôts de données de santé » correspondent aux appellations les plus couramment utilisées pour désigner les entrepôts de données de santé.

4. Commission Nationale Informatique et Libertés
5. Assistance Publique - Hôpitaux de Paris
6. Programme de Médicalisation des Systèmes d'Information

3. Traitement automatique de la langue : *natural language processing, NLP, text mining*

   Le mot clé « *text mining* » vient ici compléter le mot clé « *natural language processing* ». En effet, la fouille de textes apparait comme étant l'application du TAL la plus utilisé dans le domaine médical. C'est pourquoi le terme « *natural language processing* » peut parfois être eclipsé par le terme « *text mining* ».

À partir de ces listes de mots clés, plusieurs requêtes, présentées dans la section suivante, ont été faites sur les différents moteurs de recherche.

4 Collecte des publications

Trois moteurs de recherche bibliographiques ont été utilisés pour cette étude : PubMed, ACL Anthology et Google Scholar. PubMed est spécialisé dans la médecine et la biologie, son mode de requêtage permet de construire des requêtes qui s'appuient à la fois sur des descripteurs MeSH (*Medical Subject Headings*) et sur du langage naturel. ACL Anthology couvre la bibliographie liée à la linguistique informatique et au traitement automatique des langues. Le moteur de recherche ACL Anthology fonctionne avec Google Custom Search [7]. Google Scholar, quant à lui, n'a pas de domaine de spécialité particulier pour les publications qu'il référence. Toutes les requêtes présentées dans cette section ont été exécutées le 18 mai 2021. Les publications PudMed et ACL Anthology ont été récupérées après exécution manuelle des requêtes sur les sites web respectifs de ces bases de données bibliographiques. Quant aux publications Google Scholar, elles ont été collectées à l'aide du logiciel libre « Publish or Perish » [8].

PubMed est le moteur de recherche le plus sophistiqué parmi les trois utilisés. Nous avons donc pu croiser facilement les trois thématiques présentées précédemment et construire les requêtes suivantes :

- Requête 1 - Informatisation des données de santé

"electronic health records"[MeSH Terms] OR ("electronic health record") OR ("electronic medical record") OR ("EMR") OR ("EHR") OR ("real world data") OR ("real world evidence")

- Requête 2 - Entrepôts de données de santé

("data warehousing"[MeSH Terms] OR ("data warehouse")) AND (("clinical") OR ("biomedical") OR ("health")))

- Requête 3 - Informatisation des données de santé et Traitement automatique de la langue

(("electronic health records"[MeSH Terms] OR ("electronic health record") OR ("electronic medical record") OR ("EMR") OR ("EHR")
AND
(("natural language processing") OR ("NLP") OR ("text mining"))

- Requête 4 - Entrepôts de données de santé et Traitement automatique de la langue

("data warehousing"[MeSH Terms] OR ("data warehouse")) AND (("clinical") OR ("biomedical") OR ("health")))

7. https://developers.google.com/custom-search/
8. https://harzing.com/resources/publish-or-perish

AND
(("natural language processing") OR ("NLP") OR ("text mining"))

Pour les moteurs de recherche Google Scholar et ACL Anthology, qui ne proposent pas de requêtage avancé comme PubMed, il était plus difficile de combiner les mots clés et croiser les thématiques. De ce fait, nous nous sommes concentrés sur l'intersection entre les thématiques d'entrepôts de données biomédicales et de traitement automatique de la langue, ce qui équivaut à la requête 4 faite sur PubMed.

Pour le moteur de recherche ACL Anthology, trois requêtes ont été construites. ACL Anthology ayant un domaine de base bibliographique couvrant le TAL, les mots clés liés à cette thématique n'ont pas été pris en compte pour les requêtes. Les résultats de ces requêtes ont été concaténés et les doublons de publications ont été filtrés :

- Requête 1 - *"clinical data warehouse"*

- Requête 2 - *"health data warehouse"*

- Requête 3 - *"biomedical data warehouse"*

Pour le moteur de recherche Google Scholar, trois requêtes ont été construites. Pour chacune de ces requêtes, nous lançons une requête similaire en remplaçant « *natural language processing* » par son acronyme « *nlp* ». Les résultats de ces requêtes ont été concaténés et les doublons de publications ont été filtrés :

- Requête 1 - *"clinical data warehouse" "natural language processing"*

- Requête 2 - *"biomedical data warehouse" "natural language processing"*

- Requête 3 - *"health data warehouse" "natural language processing"*

5 Classification des publications

Les requêtes croisant les thématiques « entrepôts de données biomédicales » et « traitement automatique de la langue » ont fait l'objet d'une analyse plus approfondie grâce à revue manuelle de publications. Les publications retenues pour cette revue ont été classifiées en cherchant à répondre à 6 questions :

1. *Quel est le sujet principal de la publication ?* :
   — extraction d'informations : un des objectif de la publication est d'extraire des informations dans des données textuelles, souvent avec l'utilisation du TAL.
   — exploitation des données d'un entrepôt : utilisation des données d'un entrepôt pour une étude précise (souvent médicale).

— revue de la littérature : articles de revue de la littérature sur des thématiques précise
— présentation d'outils : articles de présentation d'outils commerciaux ou libre de droit, ces outils proposent généralement des solutions d'entreposage de données ou des applications de techniques de TAL.
— autres : publications dont le nombre de publications par catégorie est trop faible pour constituer une catégorie à part entière. Les sujets abordés dans ces publications : techniques d'entreposage de données (construction entrepôt de données, intégration de données, etc.), requêtage des entrepôts.

2. *Quel est le cas d'usage de l'entrepôt de données dans la publication ?* :
   — exploitation de données structurées
   — exploitation de données non structurées
   — construction d'entrepôts de données
   — structuration de données : structuration des données non structurées en des formats de données existants, basés sur des lexiques et/ou ontologies.
   — autres (nombre d'occurrences trop faible pour en faire une catégorie à part entière)

3. *Est-ce qu'au moins une méthode TAL est mentionné dans la publication ? Si oui, quel(s) type(s) de méthode(s) ?* :
   — linguistique
   — apprentissage automatique
   — apprentissage profond
   — inconnu : l'utilisation du TAL est mentionné mais la méthode n'est pas précisée.

4. *Quelle est la langue des données exploitées ?* (si une méthode TAL est mentionnée dans la publication)

5. *Quel est l'objectif médical dans la publication ?* (si une méthode TAL est mentionnée dans la publication) :
   — Médecine interventionnelle : étude d'un acte fort de médecine (opérations, traitements, etc.)
   — Médecine de spécialité : étude d'une maladie dans son ensemble

6. *À quelle spécialité médicale se rattachent les données exploitées ?* (si une méthode TAL est mentionnée dans la publication) : neurologie, oncologie, pneumologie, etc.

Pour la question 1, les sujets ont été obtenus de manière itérative lors de la revue manuelle des publications. Lorsqu'une publication ne pouvait être associée à un sujet existant, un nouveau sujet était créé. Sur les questions 2 et 3, les publications peuvent recevoir plusieurs réponses.

6 Analyse des publications

Depuis l'informatisation des données de santé, les données issues du soin des patients sont davantage utilisées pour la recherche clinique. La distribution des résultats obtenus avec la première requête PubMed (cf. figure 1) montre que les articles mentionnant des dossiers patient informatisés dans la littérature ont augmenté de manière exponentielle ces dix dernières années, passant 1 850 mentions en 2010 à 7 915 en 2020.

La croissance d'utilisation des entrepôts de données biomédicales se reflète également dans la littérature scientifique (cf. figure 2).

La croissance des données de santé informatisées et l'utilisation des entrepôts de données figurent parmi les facteurs qui favorisent l'usage du TAL (cf. figures 3, 4 et 5), que ce soit pour extraire de l'information, ou pour pré-traiter des données textuelles. Le TAL permet de rendre plus accessible des informations uniquement présentes dans le texte. Ces informations peuvent ensuite être utilisées dans des études de recherche clinique, ou plus généralement, ajoutées dans les entrepôts de données biomédicales afin de les enrichir. L'extraction d'informations à l'aide du TAL peut aussi permettre de récupérer des données déjà présentes de manière structurée dans les entrepôts, afin de consolider ces données, mais également de compléter les données manquantes pour certains patients.

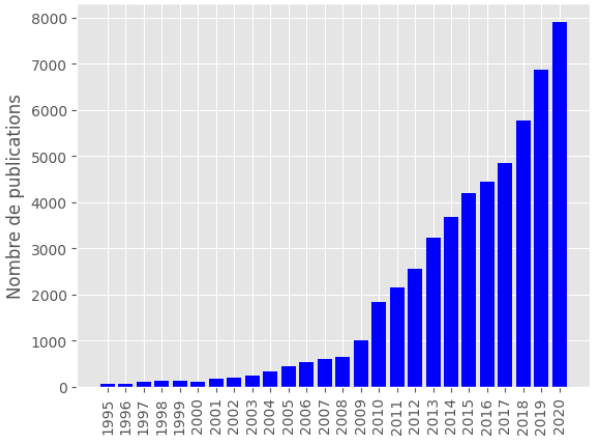


FIGURE 1 – Requête 1 - PubMed - Informatisation des données de santé

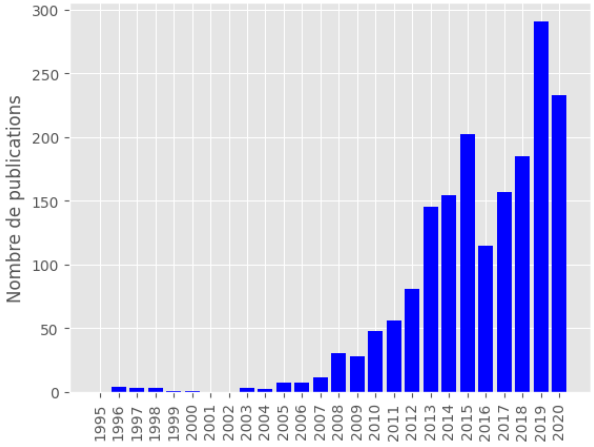


FIGURE 3 – Requête 3 - PubMed - Informatisation des données de santé et TAL

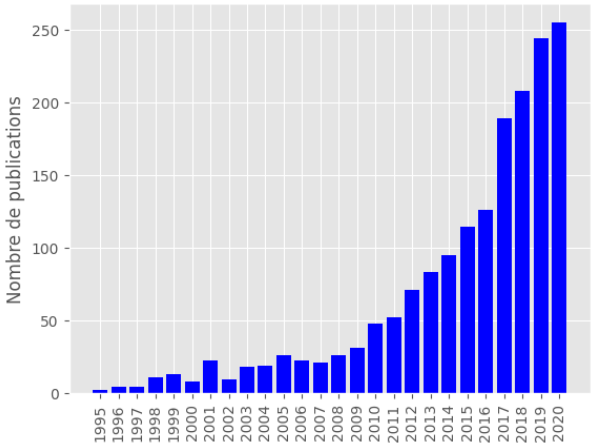


FIGURE 2 – Requête 2 - PubMed - Entrepôts de données de santé

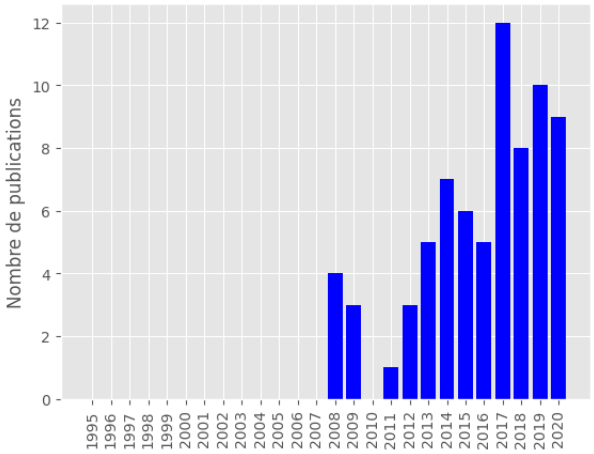


FIGURE 4 – Requête 4 - PubMed - Entrepôts de données de santé et TAL

Les résultats des requêtes croisant les thématiques « entrepôts de données biomédicales » et « traitement automatique de la langue » comptent 69 publications sur PubMed, 918 publications sur Google Scholar et seulement 3 publications pour ACL Anthology. Les publications issues de PubMed se trouvant également dans les résultats des requêtes Google Scholar ont été supprimées des résultats Google Scholar (11 publications Pubmed). Un échantillon de 80 publications de la requête Google Scholar ainsi que les 69 publications de la requête 4 PubMed ont été manuellement analysées, tandis que les publications de la requête ACL Anthology ont été abandonnées. Elles étaient trop peu nombreuses (seulement 3 publications) pour pouvoir être comparées avec les publications des autres moteurs de

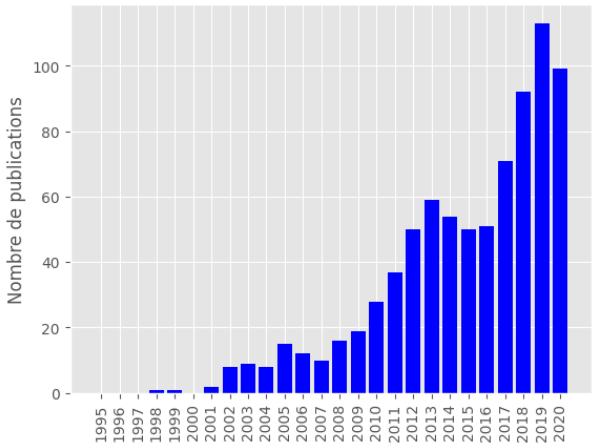


FIGURE 5 – Requête Google Scholar - Entrepôts de données de santé et TAL

recherches. Ce manque de publications sur ACL Anthology peut s'expliquer par le fait que les mots clés choisis soient trop stricts. Les publications présentes sur la base bibliographique ACL Anthology mettent généralement en avant des méthodes TAL. L'origine des données a donc moins d'importance dans ces publications, et la notion d'entrepôts de données peut paraître éloignée pour les auteurs. Les sujets traités dans ces publications sont variés (cf. figure 6). L'extraction d'informations est la thématique dominante parmi toutes ces publications puisqu'elle figure dans 74 publications (soit environ 50 % des publications revues manuellement). La thématique d'exploitation d'entrepôt de données est uniquement présente dans les publications PubMed (6 publications). Le manque d'articles sur cette thématique dans Google Scholar peut s'expliquer par le fait que cette thématique est proche du domaine médical, puisque cela correspond aux études sur données de santé. Par conséquent, on retrouve ces publications plus facilement sur Pubmed que sur Google Scholar. Parmi les publications résultants de cette requête se trouve également 33 revues de la littérature, dont la majorité provient de Google Scholar. Ces revues de la littérature portent sur différents sujets : le *big data* (Singh, 2019; Schoenthaler *et al.*, 2019), le traitement automatique de la langue (Sheikhalishahi *et al.*, 2019; Névéol *et al.*, 2018) ou plus généralement les données de santé informatisées (Safran, 2017). D'autres publications présentent différents outils ou logiciels prêts à l'emploi, tels que des outils d'entreposage de données ou d'extraction d'informations.

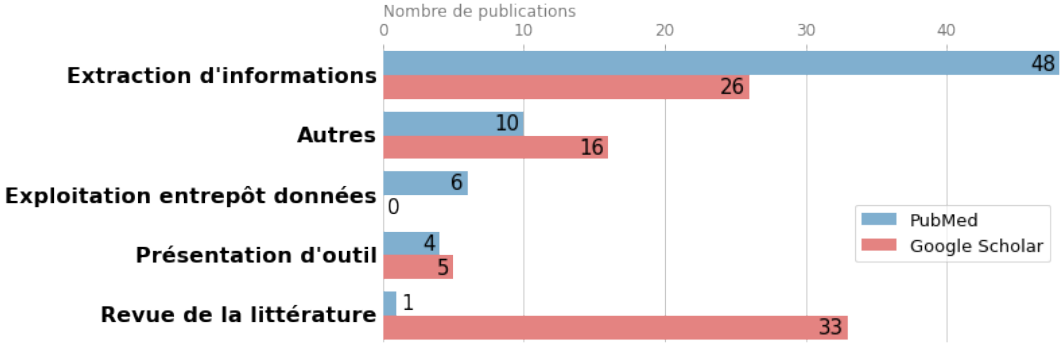


FIGURE 6 – Sujets traités dans les publications « Entrepôts de données de santé et TAL »

Les entrepôts de données peuvent avoir différents rôles (cf. figure 7). L'exploitation d'entrepôts de données est le cas d'usage le plus fréquent, avec d'un côté l'exploitation des données non structurées, présent dans 57 publications et, de l'autre, l'exploitation des données structurées, présent dans

20 publications. En amont de l'exploitation des entrepôts, la conception des entrepôts de données est également importante, avec la définition des données et des architectures qui composeront ces entrepôts. Entre ces deux tâches de conception et d'exploitation se place l'amélioration des entrepôts, avec notamment la structuration des données (Thoroddsen *et al.*, 2017; Chiudinelli *et al.*, 2019; Afshar *et al.*, 2019) déjà présentes dans l'entrepôt mais encore l'intégration de nouveaux flux de données (Delamarre *et al.*, 2015; Hernandez *et al.*, 2009).

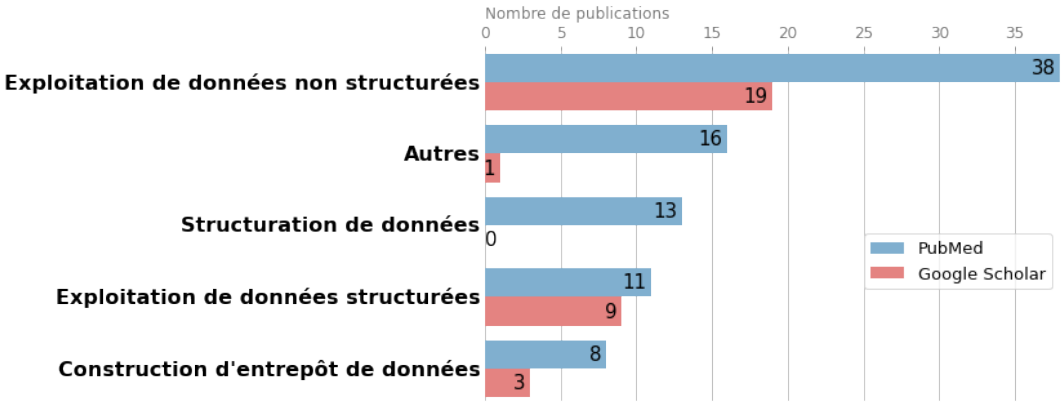


FIGURE 7 – Cas d'usage des entrepôts de données biomédicales dans les publications « Entrepôts de données de santé et TAL »

L'engouement de ces dernières années autour des méthodes à base d'apprentissage se reflète dans la littérature, la majorité des articles exploitant ces méthodes ont été publiés entre 2016 et 2020 (cf. figures 9, 10, 11 et 12). La régression (Quéroué *et al.*, 2019) et la classification (Osborne *et al.*, 2016; Chase *et al.*, 2017) comptent parmi les méthodes d'apprentissage automatique utilisées, tandis que les méthodes d'apprentissage profond s'appuient sur des réseaux de neurones (Zhao *et al.*, 2019; He *et al.*, 2019; Neuraz *et al.*, 2020). Malgré l'intérêt porté à ces méthodes, les méthodes linguistiques restent les approches les plus courantes dans la littérature médicale (cf. figures 8, 9 et 10). Parmi les méthodes linguistiques utilisées, on peut citer les approches à base de règles (Upadhyaya *et al.*, 2017; Lee *et al.*, 2020; Ryu & Zimolzak, 2020; Luther *et al.*, 2017), les expressions régulières (Wang *et al.*, 2019; Glaser *et al.*, 2018; Atti *et al.*, 2020; Kim *et al.*, 2017), ou encore les approches s'appuyant sur des lexiques (Campillo-Gimenez *et al.*, 2013; Lowe *et al.*, 2009; Evans *et al.*, 2016). Un pic de publications en 2017 portant sur les méthodes linguistiques. L'analyse des publications en question n'a pas permis d'expliquer ce pic.

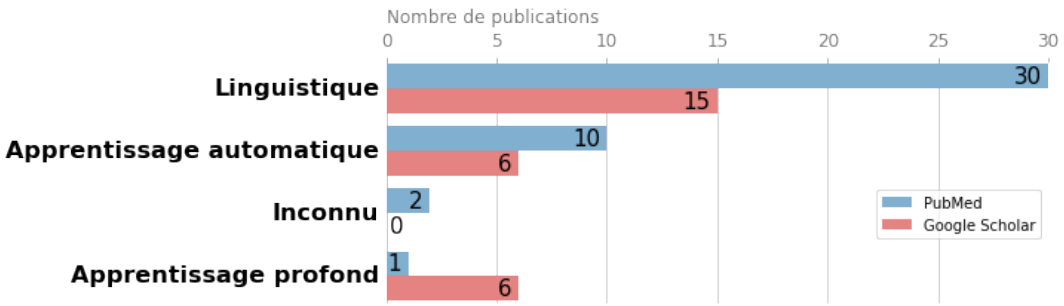


FIGURE 8 – Méthodes TAL présentes dans les publications « Entrepôts de données de santé et TAL »

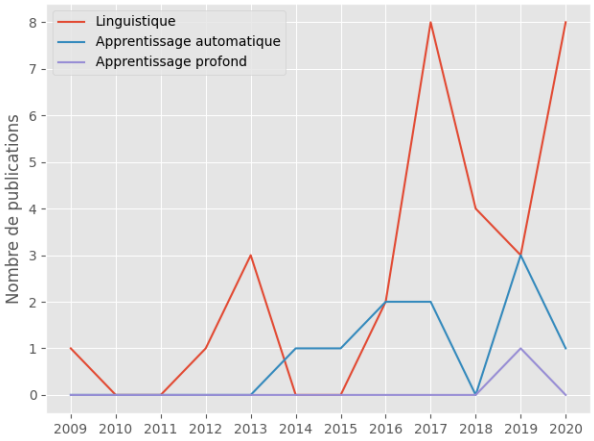


FIGURE 9 – Méthodes TAL par année des publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL sur PubMed

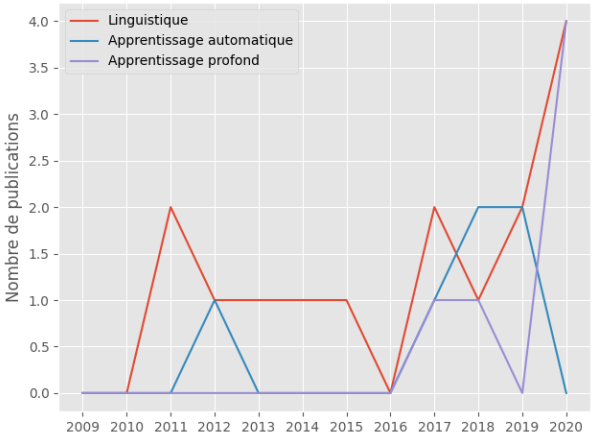


FIGURE 10 – Méthodes TAL par année des publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL sur Google Scholar

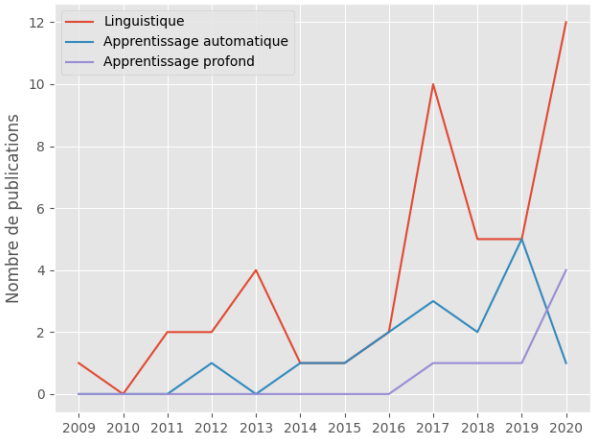


FIGURE 11 – Méthodes TAL par année des publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL (Google Scholar et PubMed cumulés)

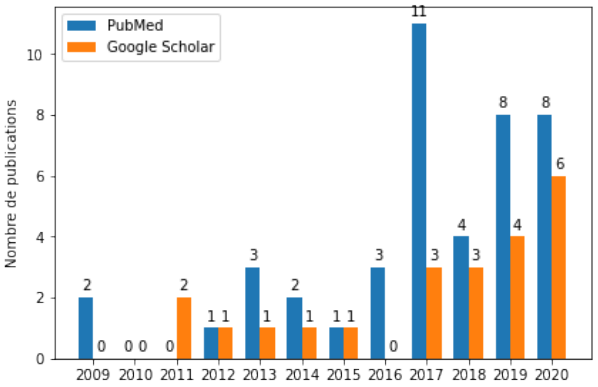


FIGURE 12 – Années de publication des publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL sur Google Scholar et PubMed

Les précédentes méthodes sont appliquées à des données médicales de différentes langues (cf. figure 13), avec une sur-représentation de la langue anglaise, mais aussi à diverses spécialités médicales (cf. figure 14). L'oncologie est la spécialité la plus traitée, suivie par la cardiologie et la neurologie. La modalité « Autres » rassemblent les spécialités médicales qui correspondent qu'à une seule publication. Parmi ces spécialités médicales, on peut retrouver la génomique, la psychiatrie, la radiologie ou encore l'endocrinologie.

L'objectif médical des études qui appliquent les méthodes de TAL peut être varié. Certaines publications portent sur la médecine interventionnelle, elles cherchent à améliorer les pratiques médicales liées à des actes forts lors de la prise en charge de patients (opérations, traitements, prélèvements biologiques, etc.). D'autres publications s'intéressent à l'étude de maladies ou de pathologies dans leur ensemble et sont classées comme médecine de spécialité. Les publications notées comme non classées traitent globalement de tâches de TAL sur des problématiques autres que le médical. L'aspect

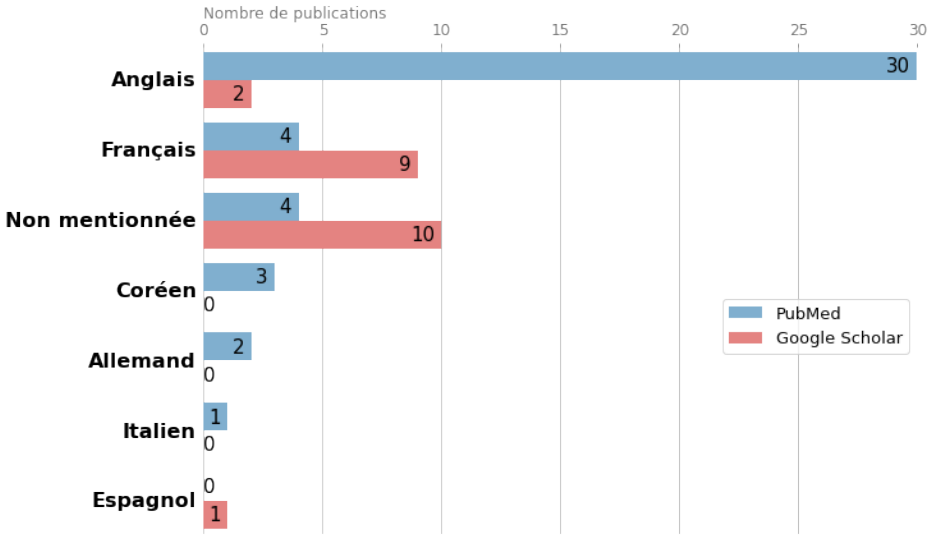


FIGURE 13 – Langue des données exploitées dans les publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL

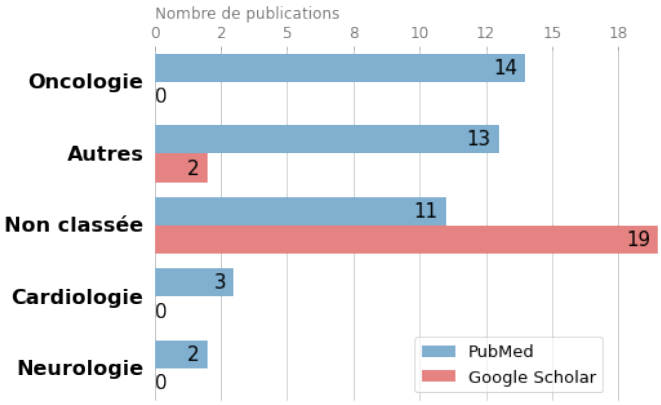


FIGURE 14 – Spécialité médicale des données exploitées dans les publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL

médical est présent dans ces publications, mais au second plan. C'est le cas pour la majorité des publications analysées qui ont été extraites de Google Scholar.

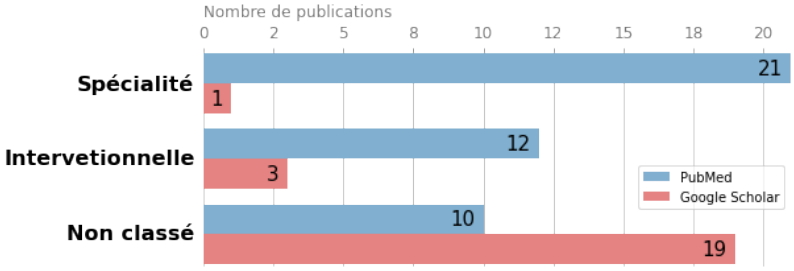


FIGURE 15 – Objectifs médicaux des publications « Entrepôts de données de santé et TAL » mentionnant une méthode TAL

7 Discussion et Conclusion

Cette revue a montré l'intérêt croissant porté aux données de santé informatisées dans la littérature biomédicale et la grande hétérogénéité des abords du TAL dans les publications. Les entrepôts de données sont au cœur de l'exploitation de ces données à des fins de recherche. Le panel de méthodes appliquées aux données textuelles médicales dans la littérature exploite bien le potentiel du traitement automatique de la langue. De plus en plus d'articles sur ces thématiques sont publiés, et ce, dans tous les champs de la santé. Sans surprise, en ayant recours à plusieurs moteur de recherche, nous avons pu également remarquer que PubMed répertorie principalement les publications où l'aspect médical est au premier plan. Les publications où les problématiques sont liés aux méthodes de TAL figurent peu sur PubMed, malgré le contexte médical présent dans ces publications. L'engouement autour du TAL et de la Santé, que l'on retrouve notamment dans le dernier numéro de la revue TAL [9], montre qu'il y a de l'intérêt pour accéder aux connaissances des données médicales, bien que l'accès à ces données soit parfois la première difficulté. Le développement du TAL dans le domaine médicale passera assurément par bien une coopération entre les experts du domaine de la santé et experts du TAL.

Remerciements

Ce travail a reçu le soutien du projet AIBy4 [10]. Nous tenons aussi à remercier les relecteurs anonymes pour leurs conseils avisés sur ce travail.

Références

AFSHAR M., DLIGACH D., SHARMA B., CAI X., BOYDA J., BIRCH S., VALDEZ D., ZELISKO S., JOYCE C., MODAVE F. & PRICE R. (2019). Development and application of a high throughput natural language processing architecture to convert all clinical documents in a clinical data warehouse into standardized medical vocabularies. *Journal of the American Medical Informatics Association*, **26**(11), 1364–1369. DOI : 10.1093/jamia/ocz068.

ATTI M., PECORARO F., PIGA S., LUZI D. & RAPONI M. (2020). Developing a surgical site infection surveillance system based on hospital unstructured clinical notes and text mining. *Surgical Infections*, **21**. DOI : 10.1089/sur.2019.238.

CAMPILLO-GIMENEZ B., GARCELON N., JARNO P., CHAPPLAIN J. & CUGGIA M. (2013). Full-text automated detection of surgical site infections secondary to neurosurgery in Rennes, france. *Studies in health technology and informatics*, **192**, 572–5. DOI : 10.3233/978-1-61499-289-9-572.

CHASE H. S., MITRANI L. R., LU G. G. & FULGIERI D. J. (2017). Early recognition of multiple sclerosis using natural language processing of the electronic health record. *BMC medical informatics and decision making*, **17**(1), 24–24. 28241760[pmid], DOI : 10.1186/s12911-017-0418-4.

CHIUDINELLI L., GABETTA M., CENTORRINO G., VIANI N., TASCA C., ZAMBELLI A., BUCALO M., GHIRARDI A., BARBARINI N., SFREDDO E., TONDINI C., BELLAZZI R. & SACCHI L. (2019).

9. https://www.atala.org/content/traitement-automatique-des-langues-et-santé
10. https://aiby4.ls2n.fr/

Ontology-driven real world evidence extraction from clinical narratives. *Studies in health technology and informatics*, **264**, 1441–1442. DOI : 10.3233/SHTI190474.

DAILLE B. & NAZARENKO A. (2017). Le tournant des données en traitement automatique des langues. . In M. BOUZEGHOUD & R. MOSSERI., Éds., *Les Big Data à découvert*, p. 118–119. CNRS editions. HAL : hal-01693019.

DELAMARRE D., BOUZILLE G., DALLEAU K., COURTEL D. & CUGGIA M. (2015). Semantic integration of medication data into the EHOP clinical data warehouse. *Studies in health technology and informatics*, **210**, 702–6. DOI : 10.3233/978-1-61499-512-8-702.

EVANS R. S., BENUZILLO J., HORNE B., LLOYD J., BRADSHAW A., BUDGE D., RASMUSSON K., ROBERTS C., BUCKWAY J., GEER N., GARRETT T. & LAPPÉ D. (2016). Automated identification and predictive tools to help identify high-risk heart failure patients : Pilot evaluation. *Journal of the American Medical Informatics Association*, **23**, ocv197. DOI : 10.1093/jamia/ocv197.

GLASER A., JORDAN B., COHEN J., DESAI A., SILBERMAN P. & MEEKS J. (2018). Automated extraction of grade, stage, and quality information from transurethral resection of bladder tumor pathology reports using natural language processing. *JCO Clinical Cancer Informatics*, **2**, 1–8. DOI : 10.1200/CCI.17.00128.

HE T., PUPPALA M., EZEANA C., HUANG Y.-S., CHOU P.-H., YU X., CHEN S., WWANG L., YIN Z., DANFORTH R., ENSOR J., CHANG J., PATEL T. & WONG S. (2019). A deep learning–based decision support tool for precision risk assessment of breast cancer. *JCO Clinical Cancer Informatics*, **3**, 1–12. DOI : 10.1200/CCI.18.00121.

HERNANDEZ P., PODCHIYSKA T., WEBER S., FERRIS T. & LOWE H. (2009). Automated mapping of pharmacy orders from two electronic health record systems to rxnorm within the stride clinical data warehouse. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, **2009**, 244–8.

KIM Y., YOON D., BYUN J., PARK H., LEE A., KIM I., LEE S., LIM H.-S. & PARK R. W. (2017). Extracting information from free-text electronic patient records to identify practice-based evidence of the performance of coronary stents. *PLOS ONE*, **12**, e0182889. DOI : 10.1371/journal.pone.0182889.

LEE K. H., KIM H. J., KIM Y.-J., KIM J. & SONG E. (2020). Extracting structured genotype information from free-text hla reports using a rule-based approach. *Journal of Korean Medical Science*, **35**. DOI : 10.3346/jkms.2020.35.e78.

LOWE H., HUANG Y. & REGULA D. (2009). Using a statistical natural language parser augmented with the umls specialist lexicon to assign snomed ct codes to anatomic sites and pathologic diagnoses in full text pathology reports. *AMIA Annual Symposium proceedings*, **2009**, 386–90.

LUTHER S. L., THOMASON S. S., SABHARWAL S., FINCH D. K., MCCART J., TOYINBO P., BOUAYAD L., MATHENY M. E., GOBBEL G. T. & POWELL-COPE G. (2017). Leveraging electronic health care record information to measure pressure ulcer risk in veterans with spinal cord injury : A longitudinal study protocol. *JMIR research protocols*, **6**(1), e3–e3. 28104580[pmid], DOI : 10.2196/resprot.5948.

MOORE N., BLIN P., LASSALLE R., THURIN N., BOSCO-LEVY P. & DROZ C. (2021). *National Health Insurance Claims Database in France (SNIRAM), Système Nationale des Données de Santé (SNDS) and Health Data Hub (HDH)*, In M. STURKENBOOM & T. SCHINK, Éds., *Databases for Pharmacoepidemiological Research*, p. 131–140. Springer International Publishing : Cham. DOI : 10.1007/978-3-030-51455-6_10.

NEURAZ A., LERNER I., DIGAN W., PARIS N., TSOPRA R., ROGIER A., BAUDOIN D., COHEN K., BURGUN A., GARCELON N. & RANCE B. (2020). Natural language processing for rapid response to emergent diseases : Case study of calcium channel blockers and hypertension in the covid-19 pandemic. *Journal of Medical Internet Research*, **22**, e20773. DOI : 10.2196/20773.

NÉVÉOL A., ZWEIGENBAUM P. & ON CLINICAL NATURAL LANGUAGE PROCESSING S. E. F. T. I. Y. S. (2018). Expanding the diversity of texts and applications : Findings from the section on clinical natural language processing of the international medical informatics association yearbook. *Yearbook of medical informatics*, **27**(1), 193–198. 30157523[pmid], DOI : 10.1055/s-0038-1667080.

OSBORNE J., WYATT M., WESTFALL A., WILLIG J., BETHARD S. & GORDON G. (2016). Efficient identification of nationally mandated reportable cancer cases using natural language processing and machine learning. *Journal of the American Medical Informatics Association*, **23**, ocw006. DOI : 10.1093/jamia/ocw006.

QUÉROUÉ M., LASHÉRAS-BAUDUIN A., JOUHET V., THIESSARD F., VITAL J.-M., ROGUES A.-M. & COSSIN S. (2019). Automatic detection of surgical site infections from a clinical data warehouse.

RYU J. H. & ZIMOLZAK A. J. (2020). Natural language processing of serum protein electrophoresis reports in the veterans affairs health care system. *JCO Clinical Cancer Informatics*, (4), 749–756. PMID : 32813561, DOI : 10.1200/CCI.19.00167.

SAFRAN C. (2017). Update on data reuse in health care. *Yearbook of medical informatics*, **26**(1), 24–27. 29063535[pmid].

SCHOENTHALER M., BOEKER M. & HORKI P. (2019). How to compete with google and co. : big data and artificial intelligence in stones. *Current Opinion in Urology*, **29**(2).

SHEIKHALISHAHI S., MIOTTO R., DUDLEY J. T., LAVELLI A., RINALDI F. & OSMANI V. (2019). Natural language processing of clinical notes on chronic diseases : Systematic review. *JMIR medical informatics*, **7**(2), e12239–e12239. 31066697[pmid], DOI : 10.2196/12239.

SINGH S. (2019). *Big Data Meets Real World! The Use of Clinical Informatics in Biomarker Research*, p. 345–352. DOI : 10.1007/978-3-030-11446-6_29.

THORODDSEN A., GUÐJÓNSDÓTTIR H. & GUDMUNDSDOTTIR E. (2017). From capturing nursing knowledge to retrieval of data from a data warehouse.

UPADHYAYA S. G., MURPHREE JR. D. H., NGUFOR C. G., KNIGHT A. M., CRONK D. J., CIMA R. R., CURRY T. B., PATHAK J., CARTER R. E. & KOR D. J. (2017). Automated diabetes case identification using electronic health record data at a tertiary care facility. *Mayo Clinic proceedings. Innovations, quality & outcomes*, **1**(1), 100–110. 30225406[pmid], DOI : 10.1016/j.mayocpiqo.2017.04.005.

WANG Y., MEHRABI S., SOHN S., ATKINSON E., AMIN S. & LIU H. (2019). Natural language processing of radiology reports for identification of skeletal site-specific fractures. *BMC Medical Informatics and Decision Making*, **19**, 73. DOI : 10.1186/s12911-019-0780-5.

ZHAO S.-C., WEI R., XIE Y.-M., WANG L.-X., WANG Q. & YI D.-H. (2019). Analysis of qingkailing injection in treatment of combined medication features of 2 147 cases of upper respiratory tract infection. *Zhongguo Zhong yao za zhi = Zhongguo zhongyao zazhi = China journal of Chinese materia medica*, **44**, 5207–5216. DOI : 10.19540/j.cnki.cjcmm.20191115.501.

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

Traduction Assistée par Ordinateur des Langues des Signes: élaboration d'un premier prototype

Marion Kaczmarek [1], Alix Larroque[2]
(1) LISN, Rue du Belvédère, 91400 Orsay, France
(2) Télécom Paris, 19 place Marguerite Perey, 91120 Palaiseau, France
marion.kaczmarek@lisn.upsaclay.fr, alix.larroque@telecom-paris.fr

RÉSUMÉ

La demande pour du contenu traduit en LSF est croissante depuis quelques années, mais l'offre est limitée par le faible nombre de traducteurs professionnels et l'absence d'outils de traduction assistée par ordinateur (TAO) dédiés pour les langues des signes (LS). Cet article s'intéresse à l'élaboration de tels outils. Après avoir étudié les méthodes de travail des traducteurs, nous avons établi un cahier des charges afin de développer un premier logiciel de TAO pour les LS. Nous avons procédé à la conception d'un tel système en développant des prototypes dits de basse fidélité avant d'implémenter une première version de logiciel fonctionnel. Nous établissons les fonctionnalités implémentées à la date de rédaction de cet article, et évoquons les fonctionnalités restant à être implémentées. Après un test du logiciel par les traducteurs professionnels, nous pourrons ensuite procéder à l'évaluation du système, afin d'améliorer son implémentation d'après leurs retours.

ABSTRACT

Computer-assisted Translation of Sign Languages: elaborating a first prototype.
The demand for Sign Language translated content has been growing in recent years, but the answer is quite limited by the scarcity of professional translators and the lack of dedicated computer-assisted translation (CAT) tools for sign languages. This article focuses on the development of such tools. After studying the working methods of translators, we established a set of specifications in order to develop a first CAT software for sign languages. To design such a system, we developed low-fidelity prototypes before implementing a first version of functional software. We list and explain the functionalities implemented at the time of writing, and discuss other functionalities still to be implemented. After testing by professionals, we aim at improving the application according to their feedback.

MOTS-CLÉS : Traduction assistée par ordinateur – Langue des signes - IHM
KEYWORDS: Computer-assisted translation – sign languages - HCI

1 Introduction

En France, la loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées parue en 2005 reconnaît la Langue des Signes Française (LSF), comme une langue à part entière, et comme une langue de la République au même titre que le français. De cette loi devrait découler une accessibilité totale en LSF pour les personnes sourdes (accueil dans les lieux publics, messages audio diffusés, information écrite). Renforcé par l'adoption en 2008 de la

Convention sur les Droits des Personnes Handicapées par les Nations Unies, le besoin pour du contenu accessible et donc traduit en LSF est croissant. Cependant, il n'y a en France encore que peu de traducteurs professionnels, et ces derniers ne sont en rien équipés comme le sont leurs collègues des langues vocales : aucun outil actuel de traduction assistée par ordinateur n'existe pour assister la traduction des langues signées.

1.1 Traduction assistée par ordinateur

Abrégée TAO, la traduction assistée par ordinateur désigne le recours à des aides logicielles par les professionnels de la traduction. Elle n'est pas à confondre avec la traduction automatique puisqu'ici, c'est bien l'opérateur humain expert qui reste aux commandes de la production. Elle se présente sous la forme d'utilitaires informatisés que le traducteur peut consulter (bases terminologiques, lexiques, glossaires etc), ou sous la forme d'environnements de travail dit intégrés, qui regroupent en une seule application de nombreuses fonctionnalités destinées à assister la tâche de traduction. Le traitement automatique des langues (TAL) y trouve une place prépondérante puisque l'on cherche à assister, si ce n'est automatiser les tâches de la traduction. Qu'il soit d'ordre syntaxique (délimitation de phrases), sémantique (traduction automatique, correcteurs orthographiques), ou par l'extraction d'informations, l'apport du TAL dans la TAO est conséquent. Démocratisée dans les années 80, elle fait aujourd'hui partie intégrante des métiers de la traduction au point que sa maîtrise tend à devenir un pré-requis pour rejoindre le monde du travail. A cheval entre informatique et traductologie, la TAO a contribué à faire évoluer le métier de traducteur. Cependant, ce constat ne s'applique pas à toutes les langues. C'est le cas des langues signées (LS), qui ne sont guère encore outillées, et dont nous détaillons les grands principes ci-après.

1.2 Les langues signées, ou langues des signes.

Les langues des signes (LS) sont des langues naturelles et orales qui utilisent les modalités visuo-gestuelle pour transmettre du sens, au travers d'articulateurs manuels, de mouvements du corps, du regard et des expressions du visage : les LS expriment plusieurs informations simultanément. Elles font usage de références spatiales persistantes dans leur organisation, qui peuvent être réutilisées dans la suite du discours. Discours qui est d'ailleurs organisé selon un ordre propre : on commence par une description préliminaire de la scène. On situe d'abord l'événement dans le temps, puis on introduit les lieux, les personnages. L'action en elle-même n'est traitée que dans un second temps. La pensée visuelle prévaut. Ainsi, l'ordre des informations présentes dans un discours diffère entre le français écrit et la LSF.

Les LS sont des langues iconiques, c'est-à-dire que les signes sont le plus souvent inspirés de la réalité (ce que la linguistique appelle l'iconicité, Cuxac, 1993). Par extension, elles sont parfois contraintes par la réalité, notamment lorsqu'il est question de géographie, ou de topologie. En effet, le contenu en LS se doit d'être visuel, et précis lorsqu'il se réfère à des relations existantes dans la réalité physique. Par exemple, lorsque l'on signe à propos de deux villes, et de la relation géographique qui les unit, le signeur doit convenablement situer les deux villes dans l'espace l'une par rapport à l'autre, en termes de distance et d'orientation.

Bien qu'il existe plusieurs systèmes pour décrire les LS sous forme graphique (SignWriting, HamNoSys par exemple, liens dans la sitographie), ils ne sont pas suffisamment utilisés par la communauté sourde au quotidien pour pouvoir être considérés comme des formes écrites. La façon la plus courante pour garder trace de la LS est l'enregistrement vidéo.

1.3 LS et informatique

La langue des signes n'est pas étrangère aux domaines informatiques. Plusieurs études, bien que récentes, s'y sont déjà intéressées, que cela soit dans le cadre de la génération automatique des LS (développement d'avatar signant) ou de la vision par ordinateur des langues des signes (reconnaissance de signes). En termes d'avatar, on citera notamment les travaux concernant la signalisation à l'adresse des personnes sourdes et malentendantes dans les gares françaises (Paire-Ficout et al, 2013). A l'internationale, l'Université DePaul de Chicago travaille sur la question depuis plus de quinze ans, et propose aujourd'hui un des avatars les plus aboutis qui soient concernant le rendu naturel des signes (Wolfe et al, 2016). D'autres études s'intéressent à la traduction des LS, mais d'un point de vue automatique, que cela soit par l'usage de gants électroniques ou d'applications smartphone.

Avant de se lancer dans le développement d'une interface de traduction dédiée aux langues des signes, il convient de s'intéresser aux logiciels déjà existants. Cette première observation nous permettra d'identifier les points clé de la TAO, mais surtout de mieux comprendre pourquoi, en l'état, ne peuvent-ils pas supporter les langues des signes comme langue de travail. La seconde étape de cette étude préliminaire s'intéresse quant à elle au processus de traduction, du français écrit vers la LSF: quelles en sont les étapes ? Sont-elles systématiques, et produites dans un ordre total ou partiel ? Ainsi, nous pourrons ensuite réfléchir à la façon d'adapter les grands principes de la TAO tant aux particularités de la LSF qu'aux pratiques traductives.

2 État de l'art

2.1 Traduction assistée par ordinateur

Les environnements de travail intégrés (c'est à dire qui regroupe tous les outils et fonctionnalités en une seule interface) destinés à la TAO fonctionnent comme suit: le traducteur charge d'abord le texte source, qu'il doit traduire. Dans un second temps, le logiciel découpe automatiquement ce texte en unités plus petites, généralement du grain de la phrase ou de la proposition, appelées segments. A chacun de ces segments sources est associé en vis-à-vis un segment cible, initialement vide. Le traducteur procède alors à la traduction, segment par segment, en ayant éventuellement recours aux outils proposés par l'environnement.

Les logiciels actuels reposent sur trois grands principes. Le premier, une forme écrite éditable: l'intégralité du logiciel repose sur l'usage d'une forme écrite éditable. En effet, aussi triviale que puisse paraître l'observation, tout passe par l'écrit: le texte source bien entendu, mais également la traduction elle-même qui sera rendue sous forme écrite, les menus, mais également les outils d'assistance tels que les dictionnaires, les glossaires, ainsi que la mémoire de traduction.

En second, la mémoire de traduction: au fur et à mesure que le traducteur traduit les segments sources, la mémoire de traduction enregistre les paires de segments sources-cibles, c'est à dire que chaque segment d'origine est appairé avec sa traduction correspondante. Si l'un des segments sources est à nouveau rencontré, à l'identique ou sous une forme très proche, alors le logiciel suggère automatiquement la traduction précédemment mémorisée: le professionnel est alors libre d'accepter, de rejeter, ou d'accepter la suggestion avec modification. La mémoire de traduction permet de capitaliser sur le travail passé. Ces bases de données sont partageables, entre collègues

d’un même service ou même parfois fournies par les clients eux-mêmes, et permettent plus de cohérence dans le temps ainsi qu’au sein des productions d’un même service.

Enfin, cee que nous avons appelé “le principe de linéarité”: l’ordre du texte source n’est pas retravaillé. Les segments sont traduits dans l’ordre d’origine, en considérant que la concaténation des segments traduits correspond à la traduction des segments sources concaténés.

Les logiciels de TAO actuels ne peuvent pas supporter la LSF parce que ces trois points clés semblent tous poser un problème d'adaptation. En effet, les langues des signes ne disposent pas de formes écrites éditables. L’usage de la vidéo pose questions en termes d’adaptation d’une mémoire de traduction puisqu’elles ne sont pas requêtables, difficilement éditables, et plus complexes à stocker en quantité. De même qu’on ne peut coller bout à bout plusieurs extraits de vidéos différentes pour produire une traduction qualitative. Enfin, comme mentionné plus haut, le discours en LSF est organisé selon un ordre propre qui diffère de celui du français, impliquant que le principe de linéarité n’est peut-être pas valable ici. La section suivante s’intéresse aux pratiques professionnelles des traducteurs, de sorte à identifier leurs besoins ainsi que les obstacles qu’ils sont amenés à rencontrer au quotidien.

2.2 Processus de traduction français écrit-LSF

Peu importe les langues, le but premier de la traduction reste de transmettre un message. Dans le cas des langues des signes, cette transition implique également un changement de modalité: on passe au visuo-gestuel. De fait, traduire du français vers la langue des signes française suppose un passage par la pensée visuelle, c’est-à-dire mettre le sens en image. Cette étape supplémentaire, dite de déverbalisation, permet d’affranchir la traduction de l’influence des constructions de l’écrit sans tomber dans le transcodage. L’exercice de reformulation, couplé à l’usage de schémas, permet au traducteur d’extraire dans un premier temps le sens du message qu’il doit faire passer, et de le reconstruire dans un second temps dans une forme propice en langue cible. (Pierre Guitteny, 2007. D. Seleskovitch et M. Lederer, 2014). La traduction texte à signes se différencie d’emblée de la traduction texte à texte par le fait qu’elle ne se construise pas au fur et à mesure du processus.

En effet, elle commence par une phase de traitement du texte source, pour établir un “scénario” de la traduction, avant que le traducteur ne procède à se filmer d’une traite. Cette méthode implique une phase d'entraînement, et de mémorisation de la version finale par le traducteur, comme nous le verrons dans le paragraphe suivant.

Afin de déterminer plus spécifiquement les étapes de la traduction français/langue des signes française, Kaczmarek et Filhol, (2020) ont mené en parallèle deux études avec des professionnels du métier. Un brainstorming d’une part, pour les amener à réfléchir et verbaliser tant leurs besoins que les problèmes rencontrés au quotidien dans leurs pratiques professionnelles. Et des sessions de traductions filmées d’autre part, où deux binômes de traducteurs étaient filmés pendant qu’ils travaillaient sur la traduction de textes journalistiques. Ils ont ainsi pu dresser une liste des tâches inhérentes à la traduction en LS, mais également déterminer si elles étaient systématiques, et ordonnées.

La figure 1 est une synthèse des tâches identifiées en observant des traducteurs à l'œuvre. La première ligne liste les tâches identifiées, et la première colonne liste les six traductions analysées, à savoir les trois mêmes textes pour chacun des deux groupes (groupe B pour “beginner” et groupe E pour “expert”). Une case verte signifie que telle tâche a été observée durant telle traduction. On

remarque que la seule tâche systématique (et la plus chronophage) est celle de segmentation et d'ordre.

	Total time	Lexical search	Discuss signs	Map search	Def. search	Encyclo search	Picture search	Seg & order	Memory
TR 1-1 (B)	1h13	■	■	■		■	■	■	
TR 2-1 (B)	59 min	■	■			■		■	
TR 3-1 (B)	12 min	■	■	■				■	
TR 1-2 (E)	19 min		■	■			■	■	■
TR 2-2 (E)	14 min		■		■			■	■
TR 3-2 (E)	11 min			■				■	■
Total time	3h08	49 min	37 min	13 min	2 min	18 min	2 min	66 min	2 min

Figure 1: Tableau illustrant les tâches observées durant des traductions fr-LSF, Kaczmarek & Filhol, 2019

Les résultats issus du brainstorming corroborent ces observations, les professionnels faisant mention de la nécessité d'accéder à du contenu en langue des signes plus facilement, qu'il s'agisse de lexique ou de traductions déjà élaborées. Ces deux études permettent également de prendre connaissance des méthodes de travail et du matériel utilisé: sollicitation de ressources sur internet, papier-crayon, prise de notes directement sur les textes… La segmentation et la modification de l'ordre des informations par exemple, passent par du repérage dans le texte, et des listings de sections numérotées. La section suivante fait le point sur les besoins identifiés et les fonctionnalités logicielles qui pourraient leur être associées.

2.3 Cahier des charges

En nous basant sur ces observations préliminaires, nous avons pu dresser une liste de fonctionnalités qu'un logiciel de TAO à destination des traducteurs en LS pourrait intégrer. La tâche de segmentation et d'ordre étant, d'après la figure 1, systématique mais également celle qui prend le plus de temps. En regard du principe de linéarité, qui ne semble pas admis de fait en LSF, et il apparaît complexe d'envisager une fonction de segmentation automatique pour le texte source. En revanche, le travail sur l'ordre (à savoir découper le texte et replacer les informations dans l'ordre propre de la LSF) peut être assisté plus facilement, tout comme le traitement du texte source ou l'accès aux ressources. Les paragraphes ci-après détaillent le cahier des charges que nous avons dressé pour le développement d'une première interface.

Gestion du texte source: Le logiciel doit être en mesure de charger et d'afficher le texte sur lequel travaille le traducteur. Il doit pouvoir prendre en charge les formats les plus communs, ainsi qu'en conserver la mise en page. En réponse aux suggestions émises lors du brainstorming quant à la possibilité de pré-traiter le texte, l'application propose les rudiments du traitement automatique, à savoir la reconnaissance de dates ainsi que la reconnaissance d'entités nommées.

Écriture et linéarité: Pour contourner deux des problèmes majeurs liés à l'adaptation des principes de la TAO pour les LS, à savoir l'absence de forme écrite ainsi que la non-application du principe de linéarité, nous suggérons un système de blocs mobiles et hiérarchisés. Le traducteur est libre de

créer autant de blocs que désiré, sous forme d'arbre. Ces derniers peuvent être librement agencés, et réorganisés à volonté. Chaque bloc peut générer plusieurs blocs enfants, eux-mêmes librement déplaçables, sachant qu'au déplacement, un bloc parent sera toujours déplacé avec ses blocs enfants. Ce système permet de faciliter la tâche d'ordre, puisque modifiable à l'infini sans altérer le contenu des blocs, évitant ainsi de tout reprendre à zéro à chaque nouvel essai. Pour contourner l'absence de forme écrite pour les LS, ces blocs peuvent accueillir différents types de contenu: de l'écrit (input au clavier), des images, ainsi que des vidéos. Les vidéos peuvent provenir soit de liens extérieurs et être importées au sein d'un bloc, soit être directement filmées à l'aide d'un outil webcam intégré à l'application.

Pour conserver une lisibilité qu'importe le degré de profondeur de l'arbre de blocs, un encart de navigation permet de dérouler ou non chaque branche de sorte à pouvoir rejoindre aisément un nœud donné dans l'arborescence.

Modules de ressources: L'application accueille un volet d'accès à différentes ressources. Chaque onglet reprend un type de recherche: lexicale, encyclopédique, ainsi qu'une carte interactive. Chacun des onglets agrège les résultats de recherche au sein de plusieurs ressources différentes. Un historique de recherche global est disponible, de sorte à pouvoir rapidement réitérer une recherche antérieure.

Prompteur: Enfin, pour considérer le processus de traduction dans son ensemble et également assister l'étape de production filmée, nous envisageons une fonction de type prompteur à intégrer à l'interface. Celle-ci permettrait au traducteur, une fois ses blocs remplis et organisés à sa guise, d'en sélectionner plusieurs de sorte à les faire défiler selon un rythme et un mode de présentation sélectionné (défilement vertical ou diaporama).

La partie suivante s'intéresse à l'élaboration d'un prototype complet basé sur ce cahier des charges, fonctionnalité par fonctionnalité.

3 Environnement de traduction intégré

C'est en suivant le cycle de conception centrée utilisateur constitué de 5 étapes (investigation, idéation, prototypage, évaluation et production), et au travers de plusieurs itérations que nous avons donc procédé à la conception d'un logiciel de TAO destiné aux langues des signes. Une première phase d'investigation a été réalisée via les études préliminaires citées ci-avant. Le processus d'idéation quant à lui a été accompli en analysant les besoins des utilisateurs pour en retirer des exigences auxquelles devra répondre notre logiciel. La prochaine étape est celle du prototypage, que la sous-section suivante détaille.

3.1 Prototypage

Dans l'optique d'explorer différentes options de design pour illustrer les fonctionnalités découlant des exigences du cahier des charges, nous avons alors entrepris de réaliser des prototypes dits de basse fidélité. Prenant la forme de prototypes papier animés (fig. 2), ces derniers nous ont permis de concrétiser notre vision rapidement afin de pouvoir projeter et réfléchir sur le design envisagé.

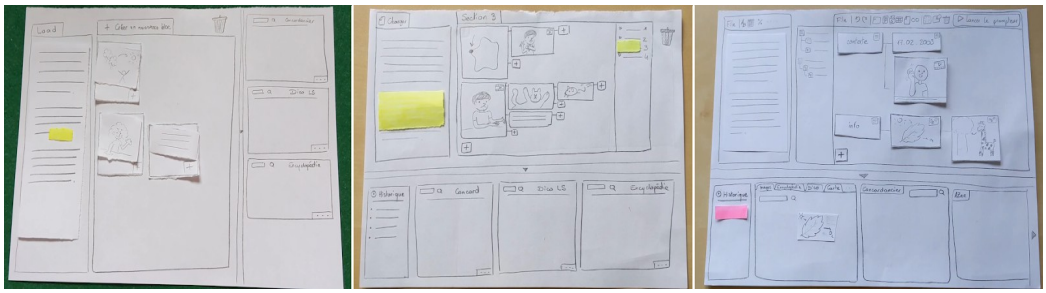


Figure 2: Premiers prototypes du système

Sur chacun de ces prototypes, nous illustrons les trois zones nécessaires à la TAO : la zone de texte source - dans laquelle on retrouvera le texte à traduire du français vers la LSF, la zone de travail principale - constituée de la zone à blocs évoquée précédemment, et enfin la zone des modules - qui constitue l'ensemble des ressources aidant à la traduction.

- Sur la gauche de l'interface, la zone de texte source permet d'afficher le texte à traduire. Des actions sont disponibles pour lancer des recherches dans les ressources depuis le texte, toujours dans l'objectif de simplifier les tâches récurrentes dans le processus de traduction (dans le cas présent, les différents types de recherche).
- Au centre de l'interface, il est possible de créer des blocs, de les remplir avec du contenu et de les déplacer suivant une structure d'arbre. Cette structure hiérarchique réorganisable répond au besoin des traducteurs de pouvoir réarranger leur discours aisément, compte tenu que cette phase de réorganisation représente 30% du temps total passé sur la traduction en plus d'être la seule tâche systématiquement observée.
- Sur la droite de l'interface, la zone des modules permet au traducteur d'accéder à différentes ressources pour l'aider dans sa traduction comme des dictionnaires en ligne ou des encyclopédies, ceci afin de répondre au besoin des utilisateurs de centraliser les ressources.

Nous avons exploré plusieurs options de design à travers chacun de ces prototypes. Après la réalisation de chaque prototype, nous avons mené plusieurs échanges avec des professionnels du métier dans le but d'obtenir des retours extérieurs sur le système, afin d'identifier des failles de design potentielles et les résoudre. En utilisant les heuristiques de Nielsen dictant les principes d'utilisabilité d'une interface homme-machine (Nielsen, 1990), nous avons ainsi procédé à plusieurs itérations successives de prototypage et d'évaluation.

En prenant l'exemple du premier prototype : après évaluation heuristique, nous avons réalisé que la position du bouton permettant de créer un bloc enfant (appelé bouton +) était visuellement peu logique (à l'intérieur du bloc parent). En l'état, le bloc enfant créé apparaît à un endroit différent du bouton ayant entraîné sa création. Cela pose un problème de visibilité de l'état du système, et c'est source potentielle de confusion pour l'utilisateur. Pour améliorer l'intuitivité du système, nous avons donc décidé de déplacer le bouton de création de bloc enfant à l'extérieur du bloc parent, et le placer à l'endroit où le bloc enfant sera créé pour une meilleure visibilité. Ce changement est visible dès le deuxième prototype.

Afin de créer un troisième prototype , nous avons demandé à une interprète en LSF de traduire deux textes en utilisant une structure de blocs pour prendre ses notes, à l'image de ce que serait capable de faire le logiciel. Cet exercice nous a apporté une nouvelle perspective, et nous a notamment permis de noter une autre faille de design présente dans le deuxième prototype. Dans ce dernier, la taille des blocs enfants est directement conditionnée à celle du bloc parent, la taille du bloc parent étant la somme de la taille de chacun de ses blocs enfants afin de montrer la hiérarchie les liant. Or, en pratique, comme on l'observe dans l'exercice réalisé par l'interprète, les blocs ayant le contenu le

plus riche - et occupant donc la plus grande place - se trouvent bien souvent à la profondeur la plus élevée. D'après ces observations et afin d'optimiser la place occupée par les blocs, il a donc été décidé de lier la taille des blocs aux contenus qu'ils abritent, et de symboliser la relation hiérarchique qui les lie à leur parent par leur position dans l'interface.

3.2 Implémentation

Une exigence importante du cahier des charges établi est de pouvoir fournir aux traducteurs un logiciel facilement accessible et ne nécessitant pas d'installation ou mise à jour avant utilisation, ceci afin de faciliter l'introduction d'un outil informatique à des utilisateurs cibles peu familiers avec ces derniers. Ainsi, implémenter le logiciel sous la forme d'une interface web permet aux utilisateurs d'y accéder en suivant un simple lien URL. Nous avons décidé de réaliser le développement avec le framework web Angular en raison de sa puissance et de sa modularité. Au terme de son développement, et une fois déployée en ligne, l'application sera gratuitement accessible.

4 Résultats

Après avoir présenté les différentes étapes de conception du système, nous pouvons nous intéresser à l'état actuel de celui-ci, après quelques mois de développement. Il a bien sûr pour vocation d'évoluer dans le temps, après de nouvelles itérations du cycle de conception, selon les retours donnés par les utilisateurs cibles notamment. La figure 3 montre l'aspect de l'interface lorsqu'un projet est en cours. L'Annexe 1 montre l'interface vide.

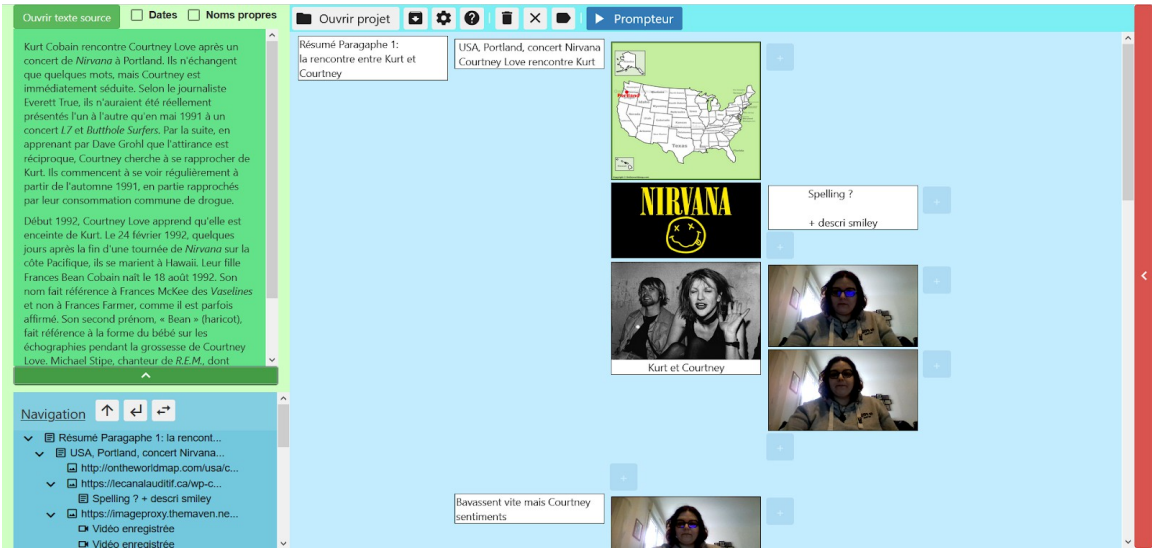


Figure 3: Capture d'écran de l'interface avec projet chargé

4.1 Liste des fonctionnalités implémentées

Voici la liste des fonctionnalités opérationnelles au moment de la rédaction:

Zone à blocs : Cœur de l'outil, la zone à blocs - au centre, en bleu sur la fig.3 - représente la zone d'édition. C'est là que l'utilisateur pourra créer de nouveaux blocs, les remplir avec du contenu de son choix, et les déplacer pour établir une hiérarchie libre à sa convenance. Cette zone permettant de prendre des notes, ainsi que d'insérer le résultat de différentes recherches réalisées au préalable par

l'utilisateur, remplit l'exigence d'avoir une structure modulaire, facilement réorganisable afin de faciliter la phase de réorganisation de la traduction, systématique et coûteuse en termes de temps (voir Annexe 4).

L'utilisateur peut créer un nouveau bloc en cliquant sur le bouton + lié au parent. Il lui est ensuite possible de le remplir avec différents contenus. Lorsqu'un bloc n'est pas encore rempli, des icônes permettant à l'utilisateur d'ajouter facilement du contenu sont affichées à l'intérieur, guidant le traducteur dans l'utilisation d'une structure peu habituelle. Ces icônes apparaissent à moitié fondues dans l'arrière-plan du bloc afin de ne pas les confondre avec du contenu réel que l'utilisateur aurait inséré par la suite (voir Annexe 3).

Actions sur les blocs : Il est possible d'ajouter du contenu au bloc sous plusieurs formes.

- Du texte, entré par l'utilisateur au clavier. Le traducteur peut ainsi entrer des notes dans des blocs, à l'image de posts-its. Couplé avec les autres types de contenu, ce texte peut également servir de légende.
- Des images, insérées via leur adresse URL ou à l'aide d'un drag and drop depuis l'image jusqu'à un bloc vide. Le traducteur peut ainsi insérer les images qu'il juge pertinentes à la préparation de sa traduction, l'aspect visuel des sujets de discussion étant primordial en LSF.
- Des vidéos, insérées depuis d'autres sites web via leur adresse URL, ou enregistrées par l'utilisateur lui-même par sa webcam, via un popup d'enregistrement dans l'outil. Cette fonctionnalité permet à l'utilisateur d'insérer des vidéos en ligne (Des définitions ou signes en LSF par exemple), ou de s'enregistrer soi-même, afin de prendre des notes directement en LSF, fonctionnalité sollicitée par les traducteurs interviewés lors des premières études.

Un seul bloc peut regrouper plusieurs contenus. Dans ces cas-là, les contenus sont réorganisables à l'intérieur d'un même bloc à l'aide d'un menu contextuel, si l'utilisateur veut par exemple afficher le texte au-dessus ou en-dessous de la vidéo. De même, il est possible pour l'utilisateur de supprimer ou de modifier individuellement chaque contenu du bloc, de supprimer tous les contenus afin de le vider tout en gardant la structure de l'arbre intacte, ou de supprimer le bloc-même ainsi que tous ses enfants. Ces deux dernières actions (suppression et vidage) impliquent l'effacement de contenus et/ou l'altération de la structure; elles sont donc accompagnées d'un avertissement pour s'assurer que leur déclenchement n'était pas accidentel.

D'autres actions sur les blocs sont possibles, comme la distribution du contenu d'un bloc parent vers des blocs enfants, ou l'inverse, laissant à l'utilisateur la possibilité de récupérer les contenus de tous les enfants et de les insérer dans le bloc parent.

Gestion du texte source : L'utilisateur peut charger un fichier .txt ou .docx. Une fois chargé, des fonctions simples de détection d'entités (noms propres et dates) sont disponibles. En effet, lors de l'observation des traducteurs professionnels, ces derniers avaient souvent le réflexe de surligner toutes les dates et/ou noms propres, et ce de manière systématique. Ces fonctions permettent donc d'automatiser la tâche. Actuellement, ces fonctions sont gérées par des expressions régulières, mais dans le futur, nous aimerions ajouter des fonctions de TAL plus poussées afin de détecter des expressions idiomatiques ou des expressions temporelles plus subtiles. (eg. "à la fin des années 80", "il y a 3 jours"...), mais aussi l'extraction automatique d'un résumé du texte source pour remplir un ou plusieurs blocs.

Modules ressources : Différents modules ressources sont accessibles depuis le logiciel, contenus dans un tiroir rétractable afin de laisser la zone principale d'édition lisible lorsque l'utilisateur n'a

plus besoin d'effectuer de recherches. Les modules actuellement disponibles sont : un module de recherche lexicale, un module de recherche encyclopédique, et un module de recherche cartographique (voir Annexe 2). Un historique commun à chacun de ces modules permet à l'utilisateur de retrouver facilement des recherches déjà effectuées, et permet de les relancer sans avoir à retaper la requête. Le module de recherche lexicale permet pour l'instant de regrouper les dictionnaires en lignes français-LSF Elix et SpreadTheSign (liens dans la sitographie). Le module de recherche encyclopédique contient Wikipédia. Dans le futur, l'utilisateur pourrait personnaliser ces modules lui-même et renseigner les sites qu'il aimerait voir accessibles. Le module de recherche cartographique contient une carte interactive sur laquelle l'utilisateur peut accrocher des épingles afin de sauvegarder certaines localisations. Il est également possible de prendre des captures d'écran pour ensuite les insérer dans des blocs de la zone d'édition. Cette fonctionnalité répond au besoin de l'utilisateur de pouvoir visualiser des lieux les uns par rapport aux autres. Des fonctions de TAL pourront ensuite permettre d'extraire les noms de lieux dans le texte source et les épingler automatiquement sur la carte dès le chargement du texte.

Fonctions système : L'export de projet au format .ZIP permet à l'utilisateur de sauvegarder l'avancement de son projet pour conservation, ou pour le partager avec des collaborateurs. Les images et vidéos générées par l'utilisateur, telles que les captures de cartes ou les enregistrements à la webcam, sont enregistrés respectivement au format .PNG et .MP4. Une simple extraction du projet zippé permet alors à l'utilisateur de récupérer ces fichiers images et vidéos si jamais il désire y accéder en dehors du logiciel. Les ressources puisées en ligne par l'utilisateur (images tirées d'encyclopédie, vidéos tirées de dictionnaires…) ne sont pas sauvegardées dans le fichier du projet afin de ne pas en augmenter inutilement la taille, seuls les liens sont conservés. Cette fonction de sauvegarde répond au besoin des utilisateurs de pouvoir différer et partager leur travail.

Un tutoriel intégré donne à l'utilisateur la possibilité de consulter à tout instant un popup expliquant les différentes parties de l'interface, afin de faciliter la prise en main du logiciel.

4.2 Liste des fonctionnalités à implémenter

Le logiciel étant toujours en cours de développement, d'autres fonctionnalités sont envisagées. Ces dernières seront le fruit de nouvelles études avec les utilisateurs cibles. Certaines sont déjà prévues :

Concordancier intégré : Part importante de tout logiciel de TAO, la mémoire de traduction que l'on souhaite conserver pour les LS prend la forme d'un concordancier bilingue (Kaczmarek & Filhol, 2020), basé sur des brèves journalistiques. Celui-ci, actuellement accessible en ligne sur une plateforme dédiée, permet à l'utilisateur de faire des requêtes de mots ou ensemble de mots afin de pouvoir visualiser les extraits en LSF (en vidéo) en contexte. L'intégrer au logiciel de TAO pour les LS permettrait ainsi l'ajout d'une ressource lexicale supplémentaire.

Prompteur : Mentionné lors de la première étude, un prompteur aiderait l'utilisateur à la production finale de la traduction. La forme exacte de celui-ci est encore indéterminée, et sa définition nécessite de nouveaux entretiens avec les utilisateurs.

L'implémentation en ligne du prototype est en cours au moment de la rédaction. Il convient dès lors de s'intéresser à son évaluation. La partie suivante détaille les pistes envisagées dans ce sens.

5 Évaluation

L'évaluation du prototype n'est pas encore réalisée au moment de la rédaction, mais elle est prévue, auprès d'un public de professionnels. Elle comprendra deux volets: une partie subjective (ou qualitative), qui a pour objectif de récolter les avis des utilisateurs, et une partie objective (quantitative), qui nous permettra d'analyser et d'évaluer les performances de l'application d'un point de vue extérieur.

5.1 Méthodologie

Un seul protocole est envisagé pour les deux types d'évaluation. Ce dernier nécessite deux groupes de professionnels : un groupe témoin, qui n'utilisera pas l'application, et un groupe test qui lui devra l'utiliser lors de la tâche proposée. Ladite tâche consiste en la traduction de plusieurs textes, du français écrit vers la LSF, et se déroule en deux parties. La première concerne les deux groupes, à qui l'on demande de traduire trois textes, avec l'aide de l'application pour le groupe test, et sans pour le groupe témoin. Les textes sont identiques entre les deux groupes, tendent vers un style journalistique et comportent deux textes dits courts (d'une dizaine de lignes, dont un de rodage) et un texte long (environ une page et demie au format portrait). Ces textes ont été choisis pour leur complexité, et les besoins de recherches annexes qu'ils pourraient susciter. La tâche de traduction va de la découverte du texte à traduire à l'étape de production filmée.

La seconde partie ne concerne que le groupe test, à qui un accès à l'application sera fourni pour une période de trois semaines. Durant cette période, le groupe test sera encouragé à réutiliser l'application dans un contexte plus libre, si possible à plusieurs reprises, sans textes imposés

5.2 Évaluation subjective

L'évaluation subjective du prototype passe par la collecte de feedback auprès des utilisateurs. Pour ce faire, nous avons établi un questionnaire de type System Usability Scale (SUS, Brooke, 1986). Il consiste en une dizaine de questions dont les réponses seront un degré attribué sur une échelle de Likert à cinq niveaux (pas du tout d'accord, pas d'accord, neutre, d'accord, tout à fait d'accord). Le questionnaire est remis aux professionnels ayant fait usage de l'application directement à la fin de la première partie. Il interroge les éléments suivants: l'aisance de la prise en main de l'application, les fonctionnalités jugées les plus utiles par le traducteur, les faiblesses de l'application, la satisfaction générale de l'utilisateur avec l'outil. Le questionnaire est également soumis aux utilisateurs au terme de la seconde partie, après trois semaines d'utilisation libre, de sorte à pouvoir comparer les résultats, et évaluer un éventuel impact de la récurrence d'utilisation.

D'autre part, nous envisageons de faire évaluer les productions des deux groupes, témoins et tests, par un troisième groupe d'experts qui lui n'aurait pas eu à effectuer la tâche de traduction. Le groupe d'experts devra attribuer une note à chaque traduction produite lors de la première partie de l'évaluation, de sorte à pouvoir analyser l'impact de l'utilisation de l'application sur la qualité de la traduction. La notion de qualité de traduction étant elle-même subjective, il reviendrait au groupe d'experts d'établir au préalable une liste de critères fixes à appliquer à chaque traduction. Pour faciliter la tâche des évaluateurs, qui devront de fait annoter des vidéos pour évaluer les traductions, nous envisageons de leur fournir un manuel d'annotations commun pour lisser leurs retours

5.3 Évaluation objective

L'évaluation objective du prototype repose elle sur des mesures indépendantes des participants. La première concerne le temps. Chaque traduction pour chaque utilisateur est chronométrée, dans le but d'établir une moyenne par texte par groupe, et de pouvoir évaluer si oui ou non l'application offre un gain de temps notable lors du processus de traduction.

D'autre part, plusieurs compteurs de clics seront implémentés dans l'application. A chaque fois qu'un utilisateur utilise telle ou telle fonctionnalité du logiciel, le compteur incrémente de un. L'analyse de ces scores permettra d'évaluer la pertinence des fonctionnalités intégrées dans l'application, fonction de sa récurrence d'utilisation (comparaison des scores entre les différentes traductions). Un écart type permettra de lisser l'influence du paramètre personnel pour les cas où une minorité d'utilisateurs auraient généré la majorité des utilisations comptabilisées.

6 Conclusion

Cet article a abordé les points clés d'une TAO adaptée pour les langues des signes. Ayant d'abord pris connaissance des réalités du métier et de ses particularités auprès des professionnels concernés, ainsi qu'ayant brossé un tableau qui résume les grands principes de la TAO, nous avons pu nous atteler à marier les deux. De l'élaboration d'un cahier des charges à l'implémentation en ligne d'une application fonctionnelle, le développement d'un tel logiciel représente un enjeu tant traductologique qu'informatique puisque les contraintes posées de part et d'autre nous incitent à repenser un domaine déjà existant. Le métier n'étant actuellement pas outillé, il est apparu complexe pour nos collaborateurs traducteurs et interprètes en langues des signes de s'imaginer quels types d'assistance un outil informatique pourrait leur apporter. De ce fait, certains choix (de design ou de fonctionnalités) lors de la conception du logiciel sont des partis pris, mais ont toujours vocation à être discutés et évalués par les professionnels concernés: il était nécessaire de trancher, et de proposer des éléments concrets pour les donner à évaluer.

Le processus de conception étant un cycle, de prochaines itérations permettront de continuer à améliorer l'ergonomie et l'utilité du système. De nouvelles perspectives issues des retours des professionnels une fois le logiciel testé, ainsi que des données collectées au travers de nouvelles études nous permettront de poursuivre le développement d'une application qui se veut issue de la collaboration entre chercheurs et professionnels de terrain. Il s'agira notamment d'inclure de nouveaux modules de ressources et de nouvelles fonctionnalités tels qu'un prompteur ainsi que l'accès direct à un concordancier bilingue en ligne. Nous envisageons également d'ouvrir une plus grande place au TAL dans notre environnement de travail intégré. Des modules empruntés au TAL des langues écrites comme évoqué précédemment, telle que la génération automatique de résumés ou encore l'extraction de pourcentages et la génération automatique de diagrammes d'une part, mais également des modules du traitement automatique des langues signées (TALS). En effet, bien que récentes, ces études s'intéressent à la formalisation des LS et à l'élaboration de modèles descriptifs pour en faciliter le traitement automatique. Des fonctionnalités comme la reconnaissance automatique de signes isolés dans une vidéo, l'aide à la segmentation du flux et le sous-titrage automatique, ou l'anonimisation des signeurs permettraient à notre environnement de travail de pouvoir également s'utiliser dans l'autre sens de langues (LSF v ers le français écrit).
Plus de contenus traduits induit une meilleure accessibilité pour les personnes signantes, ainsi qu'une visibilité accrue pour les langues des signes. Nous espérons que ces travaux trouveront une suite, et qu'ils susciteront l'intérêt de nouvelles sphères quant aux LS.

Références

BROOKE J., *(1986).* "SUS: a "quick and dirty" usability scale". *In P. W. Jordan; B. Thomas; B. A. Weerdmeester; A. L. McClelland (eds.). Usability Evaluation in Industry. London: Taylor and Francis.*

CUXAC C. (1993). « Iconicité des Langues des Signes. » *in*: *Faits de langues*, n°1, Mars 1993. Motivation et iconicité. p. 47-56.

GUITTENY P. (2007) ; « Langue des signes et schémas » *in* : *Traitement automatique des langues*, Volume 48, n°2/2007, p. 1 à 10

KACZMAREK M., FILHOL M. (2020). Alignments Data Base for a Sign Language Concordancer, *in proceedings of 12th International Conference on Language Resources and Evaluation (LREC 2020)*, p. 60696072

NIELSEN, J., MOLICH, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf. (Seattle, WA, 1–5 April),* p. 249–256

PAIRE-FICOUT L, SABY L., Alauzet A. *et al.*, « Quel format visuel adopter pour informer les sourds et malentendants dans les transports collectifs ? », *Le travail humain*, 2013/1 (Vol. 76), p. 57-78. DOI : 10.3917/th.761.0057.

SELESKOVITCH D., LEDERER M. (2014) « Interpréter pour traduire. » *5e édition revue et corrigée, 2014*, Les Belles Lettres (1re édition, 1984).

WOLFE, R., ETHIMIOUS, E, GLAUERT, J., HANKE, T., MCDONALD, J., & SCHNEPP, J. (2016) eds. Special issue: recent advances in sign language translation and avatar technology, *Springer International Publishing,* 2016

Sitographie

Elix : https://dico.elix-lsf.fr/

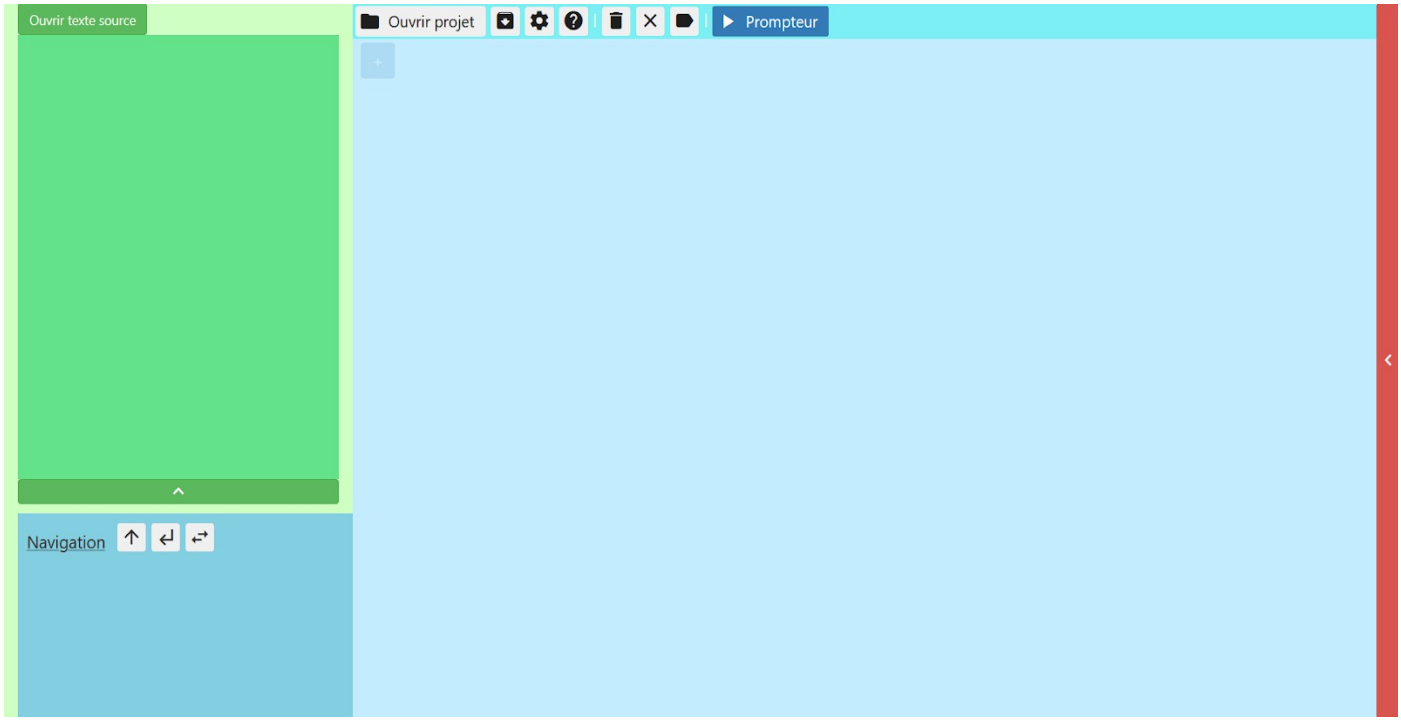
HamNoSys: http://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/hamnosys-97.html

SignWriting : https://www.signwriting.org/

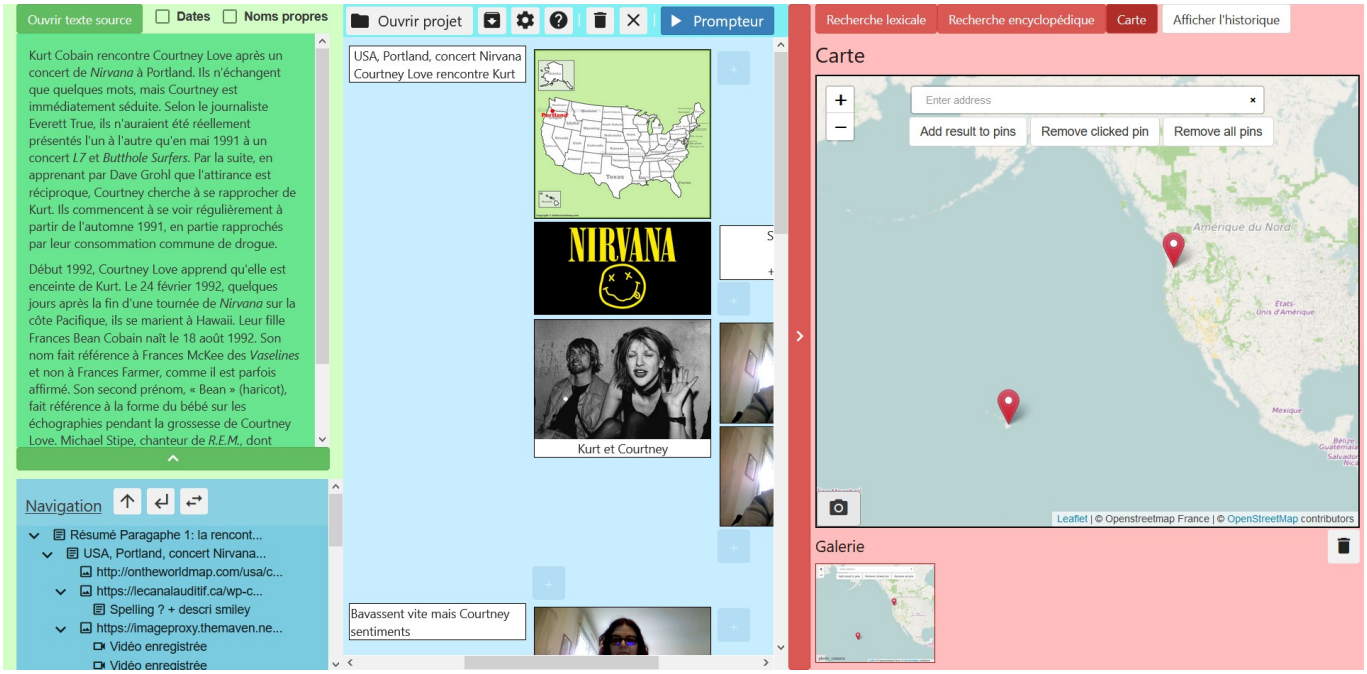
SpreadTheSign : https://www.spreadthesign.com/fr.fr/search/

Wikipédia : https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal

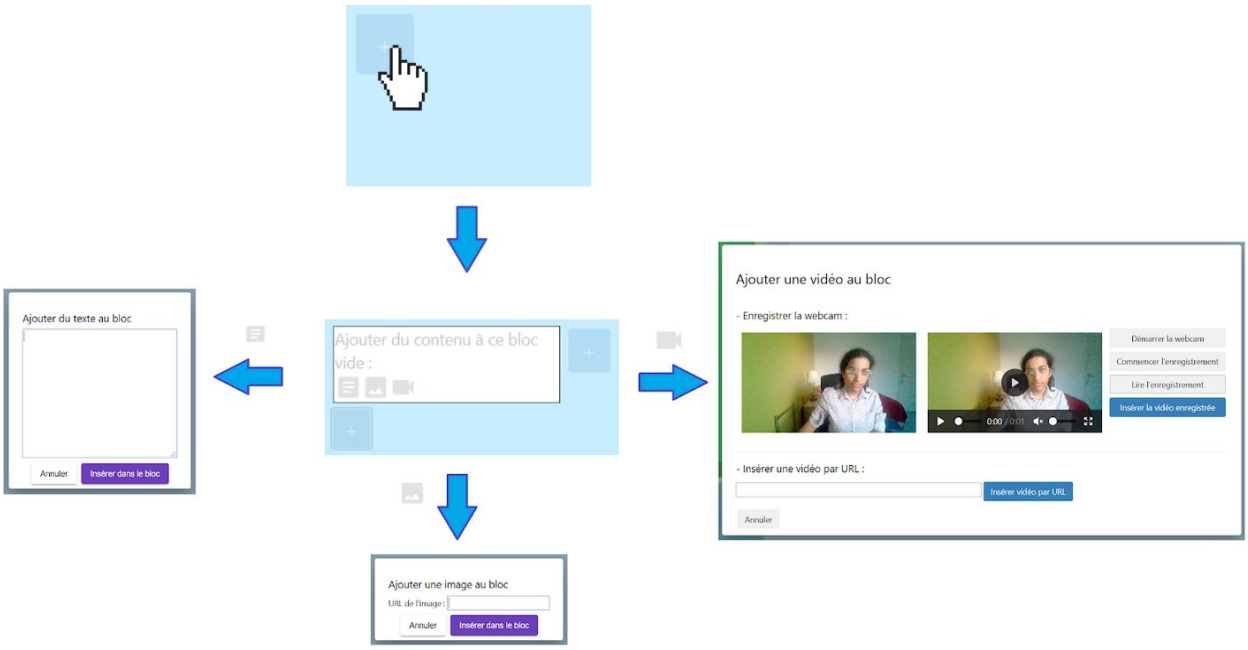
Annexe



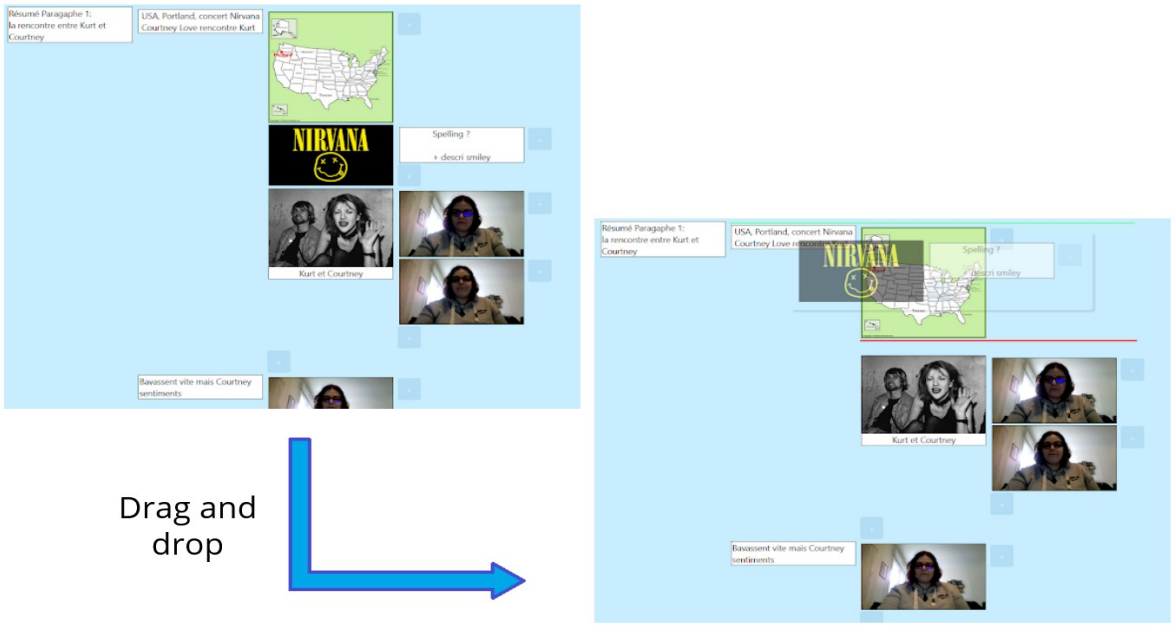
Annexe 1: Capture d'écran de l'interface à l'ouverture de la page



Annexe 2: Le module carte



Annexe 3: Différentes possibilités de contenu pour un bloc



Annexe 4: Le drag and drop permet de réagencer l'ordre des blocs

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

Utilisation d'outils de TAL pour la compréhension de spécifications de validation de données

Arthur Remaud [1,2,3]
(1) Clearsy, 320 Av. Archimède – Les Pléiades III, 13100 Aix-en-Provence, France
(2) Samovar, 9 rue Charles Fourier, 91011 Evry, France
(3) LISN, Campus Universitaire bâtiment 507, Rue du Belvédère, 91405 Orsay, France
arthur.remaud@clearsy.com

RÉSUMÉ

La validation de données consiste à vérifier formellement la cohérence de données utilisées en entrée de systèmes critiques. L'essentiel du travail des ingénieurs consiste donc à traduire une spécification, écrite en langage naturel, en un ensemble de règles formelles permettant l'automatisation de la vérification. Notre objectif à long terme est d'automatiser complètement le processus de validation de données. Dans cet article, nous présentons une première étape et détaillons les différentes techniques de traitement automatique de la langue que nous avons déployées pour générer un squelette de règle formelle à partir d'une spécification textuelle. La particularité de ces spécifications est qu'elles peuvent contenir beaucoup d'informations implicites qui rendent difficile la tâche de traduction. D'autre part, le fait qu'il n'existe pas de grand corpus d'apprentissage disponible rend difficile l'emploi des méthodes d'apprentissage neuronal profond. Néanmoins des approches plus classiques à base de règles et de représentations symboliques permettent d'apporter un premier élément de réponse.

ABSTRACT

Use of NLP tools for automatic comprehension of data validation specifications

Data validation is the formal verification of data used in critical systems. The major part of the engineers'work consists in translating a natural language specification into a set of formal rules, allowing the automation of the verification. As a first step toward a full automation of this translation process, we detail in this article different natural language processing methods which we deployed to build a formal rule skeleton from a textual specification. The characteristics of these specifications are their use of implicit project information, which make harder the task of translation. Furthermore the absence of large corpus for machine learning, makes difficult the use of deep learning neuronal methods. However, more classic approaches based on rules and symbolic representations provide a first solution.

MOTS-CLÉS : TAL, extraction d'entités, relations entre entités, analyse syntaxique.

KEYWORDS: NLP, entities extraction, entity linking, syntactic analysis.

1 Introduction

La validation formelle de données consiste en la vérification formelle des données d'entrée et de fonctionnement de systèmes critiques. L'objectif est de vérifier la cohérence des données entre elles

afin de détecter de potentielles erreurs de relevé, voire du système (Lecomte & Mottin, 2016). Par exemple, un système de guidage de train doit nécessairement avoir en entrée un plan des voies identique à la réalité. Toutes ces données sont difficiles à vérifier manuellement de par leurs tailles et nombres importants, mais on peut vérifier automatiquement qu'elles suivent les spécifications des voies, par exemple *"A signal shall be at least 100 meters before a crossover"*.

Le travail des ingénieurs en validation de données consiste en grande partie à traduire en règles formelles les spécifications des clients concernant les propriétés à vérifier sur les données. La majorité de ces spécifications sont des phrases simples, que l'on voudrait pouvoir traiter automatiquement ou semi-automatiquement, afin que les ingénieurs consacrent plus de temps aux spécifications plus complexes qui demandent plus de réflexion.

Dans cette optique, l'utilisation d'outils d'analyse de la langue est nécessaire afin de pouvoir saisir le sens de la vérification à appliquer. La traduction de spécifications en règles formelles a déjà été exploré par le passé (Sadoun, 2014), mais avec des techniques de traitement de la langue éloignées de l'état de l'art. Actuellement, les programmes les plus performants pour la traduction, et l'extraction d'informations d'un texte reposent sur des techniques d'apprentissage automatique (Devlin *et al.*, 2018; Brown *et al.*, 2020), utilisant d'importants corpus de textes pour la phase d'apprentissage. Or, les différents projets industriels de validation de données ne comprennent, en général, guère plus de quelques centaines de spécifications au maximum, avec de grandes différences d'un projet à l'autre. A la connaissance des auteurs, il n'existe pas de corpus dans ce domaine permettant d'entraîner des réseaux de neurones profonds pour des tâches de traitement automatique de la langue. A cause de la grande complexité de la production de données annotées, et des données existantes en faible quantité, il convient d'expérimenter ces réseaux neuronaux pour des tâches plus simples que la traduction directe, et de combiner ces résultats pour parvenir à cet objectif.

Cet article montre différentes approches du traitement de la langue utilisées dans cet objectif industriel, afin d'établir un prototype pour traduire des spécifications, en partant d'un modèle de phrase précis (proche d'un langage contrôlé) pour ensuite l'étendre sur des formulations plus complexes et plus variées. La section 2 détaillera les techniques abordées, tout en prenant en compte la contrainte du corpus d'apprentissage afin de concevoir le prototype. Ensuite la section 3 détaillera un essai du prototype sur des spécifications issues d'un projet industriel. Enfin la conclusion apportera un bilan de ce travail et les perspectives d'avenir pour ce prototype.

2 Développement du prototype

A l'heure actuelle, les outils les plus performants en TAL se basent sur des réseaux de neurones profonds, dont la dernière couche est adaptée à la tâche voulue dans une étape d'apprentissage appelée *fine-tunning*, comme l'approche BERT (Devlin *et al.*, 2018) à base de « Transformer » (réseau neuronal profond avec une architecture encodeur-décodeur). Nous avons donc décidé d'utiliser des outils qui s'appuient sur ces algorithmes afin d'étudier leurs performances dans le cadre de la validation de données, limité notamment par le manque de données d'entraînement. Cette utilisation sera complétée par des des outils à base de règles pour compléter une traduction en règle formelle.

Parmi les informations que nous voulons extraire, nous voulons principalement savoir quelles données vont être analysées et quelle(s) vérification(s) est (sont) à appliquer. Pour cela, l'extraction d'entités et l'étiquetage de relations permet de ressortir les données et de voir comment elles s'articulent entre

elles, et l'analyse syntaxique indiquera la vérification à effectuer et l'ordre des données en paramètre.

2.1 Extraction d'entités

Chaque spécification produit une vérification à faire sur certaines données. On retrouve donc des noms d'objets, leurs paramètres ou simplement des valeurs que l'on aimerait détecter automatiquement. Dans notre cas, on recense quatre types d'entités à extraire :

— les classes, qui représentent les objets à analyser dans les données,
— les attributs, qui sont des paramètres des classes,
— les variables,
— les valeurs, comme des entiers, des chaînes de caractères, ou des valeurs qualitatives.

Parmi les approches récentes qui utilisent des plongements lexicaux de type BERT, nous avons utilisé l'application Bert-NER [1] qui a été choisie, car elle est facile d'utilisation en tant que bibliothèque, les phases de *fine-tuning* et d'utilisation d'un modèle entraîné étant bien découplées. Bien que ce programme n'utilise pas d'aide pour les informations implicites du projet, ses résultats sont satisfaisants, comme montré plus loin dans la section 3.2.1. Il existe différents modèles pré-entraînés, mais ils ne permettent pas de détecter les entités recherchées dans les spécifications, car le langage utilisé pour les spécifications et les concepts qui y sont référencés sont spécifiques au domaine. Pour y remédier, 140 spécifications d'un projet industriel ont été annotées afin de faire un *fine-tuning* spécialisé à notre tâche.

Associées aux entités, les relations qui les lient sont une source d'informations utiles pour la traduction de spécification dans un langage formel.

2.2 Étiquetage de relation entre entités

Les entités d'un texte sont très souvent liées entre elles. De nombreux travaux en TAL concernent l'identification des liaisons entre ces entités et la détermination du type des relations, comme par exemple entre un objet et son possesseur (Shi & Lin, 2019; Papanikolaou *et al.*, 2019).

Nous cherchons à étiqueter quatre types de relations entre les entités :

— les relations classe-attribut, déterminant à quelle classe est attaché un paramètre,
— les relations variable-valeur, indiquant à quelle variable est affectée une valeur,
— les relations variable-type, reliant une variable à son type lorsqu'il est indiqué dans le texte,
— les relations valeur-paramètre, indiquant qu'une valeur (données, variable) est conditionnée par un paramètre.

Ces relations permettent entre autre de faire de l'affectation de valeur à une variable, de faire du typage, ou d'éviter des ambiguïtés comme lorsque que plusieurs classes ont un attribut de même nom. Elles ne peuvent toutes être détectées par une analyse syntaxique car ces relations peuvent concernés des mots très éloignés dans une phrase.

Pour cet objectif, nous avons repris l'outil OpenNRE (Han *et al.*, 2019) sous licence MIT. A nouveau, il a fallu faire un *fine-tuning* pour les relations précédemment citées, avec le même jeu de données que celui utilisé pour l'extraction d'entités, étiqueté pour cette tâche. L'un des avantages de cet outil est qu'avec l'étiquetage de la relation, il y a un pourcentage de certitude de l'algorithme pour aider à

1. https://github.com/kamalkraj/BERT-NER

déterminer la véracité de l'information.

Pour compléter ces informations sur les données à valider, il faut aussi les informations sur le type de vérification demandée, que l'on peut trouver avec de l'analyse syntaxique.

2.3 Analyse de l'arbre syntaxique

Pour compléter l'étude des phrases, une première approche peut être l'analyse de sa structure et sa syntaxe. En regardant comment les mots s'articulent entre eux, on peut déjà apercevoir certaines relations sémantiques, notamment pour les phrases les plus basiques.

Dans beaucoup de règles, on observe que le groupe verbal indique la vérification à effectuer, par exemple *is greater than* ou *contains*. Pour chaque groupe verbal, on peut déterminer l'arité de l'opérateur associé ainsi que les rôles du sujet et des différents compléments, qui constituent ainsi les paramètres de la vérifications. Une liste de toutes les vérifications les plus courantes dans les spécifications a été construite pour pouvoir les repérer avec l'analyse syntaxique, avec à chaque fois les traductions en langage formel.

L'analyse des paramètres de la vérification repose sur les informations syntaxiques, mais aussi sur la présence des entités et leurs relations entre elles extraites par les outils des sections précédentes. Comme pour les groupes verbaux, une liste des principaux types de sujets et compléments est dressée, avec par exemple un sujet ne contenant qu'une donnée, ou un complément composé du mot *range* explicitant l'utilisation d'un intervalle, etc.

Pour l'analyse syntaxique, c'est la bibliothèque Python *Spacy* (Honnibal & Montani, 2017) qui a été retenue pour sa simplicité d'usage et ses résultats satisfaisants.

2.4 Assemblage dans un prototype

Toutes les techniques décrites précédemment sont assemblées dans un prototype qui permet d'analyser les spécifications textuelles et construire un squelette rédigé dans un langage naturel contraint, utilisé notamment pour que le client puisse valider la règle formelle facilement.

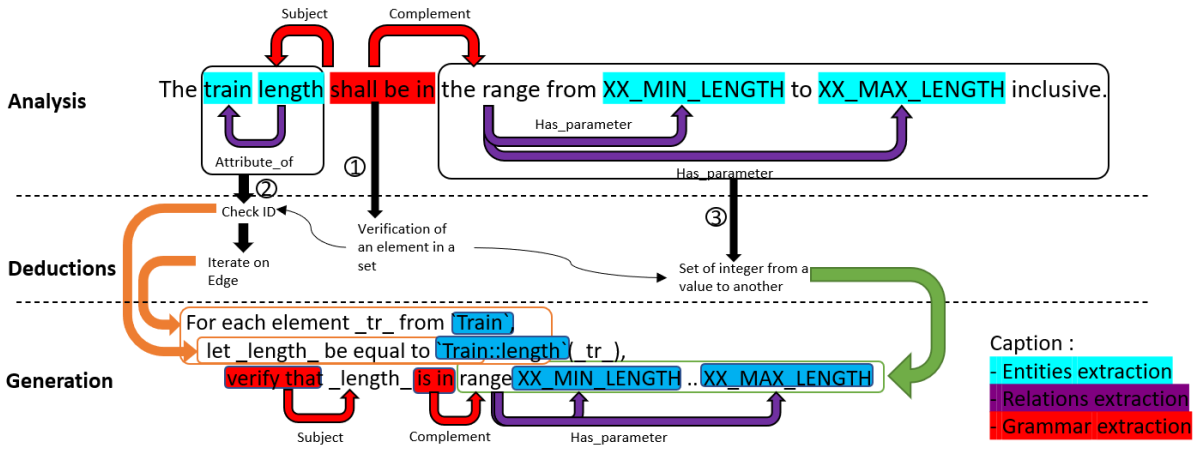


FIGURE 1 – Fonctionnement du prototype

Comme le montre la figure 1, nous utilisons l'analyse grammaticale afin d'extraire le groupe verbal (dans cet exemple *"shall be in"*) pour identifier la vérification (ici l'appartenance à un ensemble), ainsi que le sujet et les compléments pour avoir les paramètres. Dans ces paramètres, les programmes décrits dans les paragraphes précédents aidés des données du projet extraient les données (ici la taille du train), les variables (comme l'intervalle) et les valeurs (ici les constantes *XX_MIN_LENGTH* et *XX_MAX_LENGTH*), ainsi que les relations entre elles (respectivement *class-attribute* et *value-parameter*).

Chaque élément est traduit par une règle préétablie en langage naturel contraint, et le tout est assemblé en suivant l'architecture grammaticale afin d'obtenir la règle finale. Ainsi la règle itère sur la donnée indiquée par le sujet, puis s'assure que cette donnée est bien dans un intervalle entre deux constantes.

3 Utilisation du prototype sur un projet concret de validation de données

Les différentes techniques d'analyse de spécifications présentées dans la section précédente ont été testées sur un projet concret de validation de données de l'entreprise Clearsy [2]. L'objectif n'est pas de tout traduire parfaitement, mais de quantifier le nombre de spécifications pouvant être analysées entièrement par ce prototype.

3.1 Présentation du projet

Le projet est constitué de 188 spécifications rédigées en anglais, constituées d'une seule phrase, le prototype ne faisant pas le lien entre différentes phrases. Elles sont toutes classées en fonction de la difficulté du texte utilisé pour estimer la faisabilité de la traduction par le prototype. En tout, 33 sont considérées comme suffisamment basiques, avec une structure grammaticale simple et des compléments facilement identifiables pour pouvoir être traitées entièrement automatiquement sans erreur (exemple : « *For a specific zone of type DEFAULT, the area length shall be greater than MIN_SPECIFIC_ZONE_LENGTH* » [3]. Ce projet a déjà été traité manuellement, donc les résultats de ce test peuvent être comparés avec ce que les ingénieurs ont rédigé.

3.2 Analyse des résultats

3.2.1 Analyse des outils d'extraction d'informations

L'outil d'extraction d'entités, obtient des résultats très corrects. Près de 90% des entités voulues sont extraites, avec un peu plus de 10% de faux négatifs. De plus, seulement 2% des entités extraites sont des faux-positifs, ce qui fait un score F1 de 0,93.

L'étiquetage de relations n'est pas fiable. Les pourcentages de certitude de l'outil présenté ne dépassent pas les 20%, ce qui est trop faible pour différencier les vrais-positifs des faux-positifs par

2. clearsy.com
3. Les noms des données et le détail de la spécification ont été modifiées pour ne pas divulguer les informations confidentielles

un programme. Ces derniers sont donc très nombreux, par exemple dans la spécification « *The area length shall be greater than train length.* » , la relation correcte entre *area* et *length* de classe-attribut est trouvée avec une certitude de 16,4%, mais une relation fausse de même catégorie est aussi extraite entre *train* et *area* avec 13,44%, et il est difficile de trouver un seuil d'acceptation correct entre ces valeurs. Les explications possibles de ce manque de précision sont multiples, mais les principales pistes reposent sur le manque de données d'entraînement, ou un manque à l'entraînement d'entités n'ayant pas de relations entre elles. Pour compenser cela, une piste envisagée, autre que créer plus de données annotées, serait d'ajouter des informations connues du projet, sous la forme par exemple de graphes de connaissances.

La construction de l'arbre syntaxique diverge très rapidement des règles préétablies, avec des mauvaises liaisons trouvées entre les parties de la phrase, par exemple un intervalle avec les mot-clés *from* et *to* rattachés au verbe plutôt qu'au complément. Dès que les phrases emploient une structure un peu plus complexe que les schémas attendus, ces liaisons ne sont pas construites comme elles le devraient, et il faudrait à chaque fois rajouter une règle pour les traiter.

3.2.2 Analyse de la traduction

Sur les 188 spécifications :

- 6 ont été entièrement traduites, avec la bonne vérifications et les bons paramètres (3%),
- 154 sont partiellement traitées, avec une partie des paramètres ou la vérification non-reconnus (82%),
- 28 ne sont pas gérées par le prototype (15%), car reposant sur une structure grammaticale trop complexe.

Les règles entièrement traduites sont celles dérivées de l'exemple choisi comme base, présenté dans la figure 1. Seules les données changent, les formulations étant quasiment identiques.

En revanche, on s'aperçoit que le reste des spécifications sont assez mal gérées. Certaines plutôt simples grammaticalement sont quasiment entièrement traduites, mais pour la plupart il n'y a que certains morceaux qui sont reconnus, l'essentiel de la phrase restant trop complexe pour un prototype de cette portée. Tout d'abord, toutes les phrases utilisant d'autres formes syntaxiques que *sujet - groupe verbal - complément* ne sont pas traitées correctement, et pour toutes les spécifications qui s'étalent sur plusieurs phrases, il faut gérer les connexions entre celles-ci. Ensuite, même pour les phrases simples, le moindre changement dans les formulations utilisées peut faire basculer l'analyse de l'arbre syntaxique, et s'adapter à chaque situation s'avère pénible pour peu de gains.

L'analyse syntaxique montre dans cet exemple ses limitations, à savoir la rigidité en cas de légère variation. Ce prototype est peu utilisable dans un contexte industriel, mais les différents outils utilisés forment une base à améliorer pour extraire des informations plus précises.

4 Conclusion

La validation de données est une étape nécessaire pour le développement de systèmes critiques. Le principal travail des ingénieurs dans cette tâche est de traduire en règles formelles les spécifications écrites en langage naturel. Pour aider et accélérer ce processus, les outils de traitement de la langue semblent tout indiqués.

Plusieurs outils sont combinés afin d'extraire le plus d'informations possibles dans un prototype. Celui-ci associe ces informations avec des schémas existants afin de traduire automatiquement les spécifications les plus simples. Les résultats de l'utilisation de ce prototype sur un projet industriel montrent les limitations de cette approche, à savoir le manque de souplesse par rapport aux variations syntaxiques. Pour accomplir cette tâche de traduction, d'autres pistes sont envisagées, notamment du côté de l'analyse sémantique des phrases, aidées des connaissances implicites du projet via par exemple un graphe de connaissances.

Remerciements

Merci à Maximilien Colange, Catherine Dubois et Patrick Paroubek pour leur aide sur ce travail et la relecture de cet article.

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.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding.

HAN X., GAO T., YAO Y., YE D., LIU Z. & SUN M. (2019). OpenNRE : An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP : System Demonstrations*, p. 169–174.

HONNIBAL M. & MONTANI I. (2017). spaCy 2 : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

LECOMTE T. & MOTTIN E. (2016). Formal data validation in the railways.

PAPANIKOLAOU Y., ROBERTS I. & PIERLEONI A. (2019). Deep bidirectional transformers for relation extraction without supervision.

SADOUN D. (2014). *Des spécifications en langage naturel aux spécifications formelles via une ontologie comme modèle pivot*. Theses, Université Paris Sud - Paris XI.

SHI P. & LIN J. (2019). Simple BERT models for relation extraction and semantic role labeling.