Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL.

Christophe Benzitoun, Chloé Braud, Laurine Huber, David Langlois, Slim Ouni, Sylvain Pogodalla, Stéphane Schneider

6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition) (JEP-TALN-RÉCITAL) [1]

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition).

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL

Christophe Benzitoun, Chloé Braud, Laurine Huber, David Langlois, Slim Ouni, Sylvain Pogodalla, Stéphane Schneider (Éds.)

Nancy, France, 08-19 juin 2020

1. https://jep-taln2020.loria.fr/

Avec le soutien de

Message des présidents de l'AFCP et de l'ATALA

En ce printemps 2020, et les circonstances exceptionnelles qui l'accompagnent, c'est avec une émotion toute particulière que nous vous convions à la 6e édition conjointe des Journées d'Études sur la Parole (JEP), de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) et des Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL). Après une première édition commune en 2002 (à Nancy, déjà !), et une expérience renouvelée avec succès en 2004, c'est désormais tous les quatre ans (Avignon 2008, Grenoble 2012) que se répète cet événement commun, attendu de pied ferme par les membres des deux communautés scientifiques voisines.

Cette édition 2020 est exceptionnelle, puisque dans le cadre des mesures sanitaires liées à la pandémie mondiale de COVID-19 (confinement strict, puis déconfinement progressif), la conférence ne peut avoir lieu à Nancy comme initialement prévu, mais se déroule à distance, sous forme virtuelle, soutenue par les technologies de l'information et de la communication. Nous remercions ici chaleureusement les organisateurs, Christophe Benzitoun, Chloé Braud, Laurine Huber, David Langlois, Slim Ouni et Sylvain Pogodalla, qui ont dû faire preuve de souplesse, d'inventivité, de détermination, de puissance de travail, et de tant d'autres qualités encore, afin de maintenir la conférence dans ces circonstances, en proposant un format inédit. Grâce aux différentes solutions mises en œuvre dans un délai court, la publication des communications scientifiques est assurée, structurée, et les échanges scientifiques sont favorisés, même à distance.

Bien entendu, nous regrettons tous que cette réunion JEP-TALN-RECITAL ne permette pas, comme ses prédécesseurs, de nouer ou renforcer les liens sociaux entre les différents membres de nos communautés respectives – chercheurs, jeunes et moins jeunes, académiques et industriels, professionnels et étudiants – autour d'une passionnante discussion scientifique ou d'un mémorable événement social... Notre conviction est qu'il est indispensable de maintenir à l'avenir de tels lieux d'échanges dans le domaine francophone, afin bien sûr de permettre aux jeunes diplômés de venir présenter leurs travaux et poser leurs questions sans la barrière de la langue, mais aussi de dynamiser nos communautés, de renforcer les échanges et les collaborations, et d'ouvrir la discussion autour des enjeux d'avenir, qui questionnent plus que jamais la place de la science et des scientifiques dans notre société.

Lors de la précédente édition, nous nous interrogions sur les phénomènes et tendances liés à l'apprentissage profond et sur leurs impacts sur les domaines de la Parole et du TAL. Force est de constater que l'engouement pour ces approches dans nos domaines a permis un retour sur le devant de la scène des domaines liés à l'Intelligence Artificielle, animant parfois un débat tant philosophique que technique sur la place de la machine dans la société, notamment à travers le questionnement sur la vie privée de l'utilisateur. Ces questionnements impactent tant la Parole que le TAL, d'une part sur la place de la gestion des données, d'autre part sur les modèles eux-mêmes. Malgré ces questionnements, nous constatons que les acquis et les expertises perdurent, et les nouvelles approches liées à l'apprentissage profond ont permis un rapprochement des domaines de la Parole et du TAL, sans les dénaturer, à la manière des conférences JEP-TALN-RECITAL qui créent un espace plus grand d'échange et d'enrichissement réciproques.

Nous terminons ces quelques mots d'ouverture en remerciant l'ensemble des personnes qui ont rendu possible cet événement qui restera, nous l'espérons, riche et passionnant, malgré les circonstances. L'ATALA et l'AFCP tiennent tout d'abord à réitérer leurs remerciements aux organisateurs des JEP, de TALN et de RECITAL, qui sont parvenus à maintenir le cap à travers vents et marées. Nos remerciements vont également à l'ensemble des membres des comités de programme, dont le travail et l'implication ont permis de garantir la qualité et la cohérence du programme finalement retenu. Un grand merci aux relecteurs pour le temps et le soin qu'ils ont dédiés à ce travail anonyme et indispensable. Ils se reflètent dans la qualité des soumissions que chacun pourra découvrir sur le site de la conférence.

En conclusion, cette 6e édition conjointe JEP-TALN-RECITAL est exceptionnelle parce qu'elle se tient dans un contexte de crise généralisée — crise sanitaire, économique, voire sociale et politique. Mais nous

formons le vœu qu'elle reste également dans les annales pour la qualité des échanges scientifiques qu'elle aura suscités, et pour le message envoyé à nos communautés scientifiques et à la société dans son ensemble, un message de détermination et de confiance en l'avenir, où la science et les nouvelles technologies restent au service de l'humain.

Véronique Delvaux, présidente de l'Association Francophone de la Communication Parlée
Christophe Servan, président de l'Association pour le Traitement Automatique des LAngues

Préface

En 2002, l'AFCP (Association Francophone pour la Communication Parlée) et l'ATALA (Association pour le Traitement Automatique des Langues) organisèrent conjointement leurs principales conférences afin de réunir en un seul lieu, à Nancy, les communautés du traitement automatique et de la description des langues écrites, parlées et signées.
En 2020, la sixième conférence commune revient à Nancy, après Fès (2004), Avignon (2008), Grenoble (2012) et Paris (2016). Elle est organisée par le LORIA (Laboratoire lorrain de recherche en informatique et ses applications, UMR 7503), l'ATILF (Analyse et traitement informatique de la langue française, UMR 7118) et l'INIST (Institut de l'information scientifique et technique) et regroupe :

— les 33es Journées d'Études sur la Parole (JEP),
— la 27^{e} conférence sur le Traitement Automatique des Langues Naturelles (TALN),
— la 22^{e} Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL).

Les circonstances particulières liées à l'épidémie de Covid-19 en France et dans le monde ont conduit à une virtualisation de la conférence. Ainsi, malgré un rassemblement physique qui n'a pu avoir lieu, diffusions, présentations (au gré des auteurs) et discussions des articles acceptés ont lieu sur le site internet de la conférence. Les tutoriels, certains ateliers, et le salon de l'innovation qui accompagnent la conférence ont cependant dû être annulés, mais les ateliers suivants sont maintenus :

— Défi Fouille de Textes (DEFT 2020),
— Éthique et TRaitemeNt Automatique des Langues (ÉTeRNAL).

La conférence accueille également des conférencières et conférenciers invités dont les exposés sont diffusés sur le site : Dirk Hovy (université de Bocconi, Milan, Italie, invité ÉTeRNAL) ainsi que Marie-Jean Meurs (Université du Québec à Montréal, UQAM, Canada) et Hugo Cyr (Faculté de science politique et droit à l'Université du Québec à Montréaln UQAM, Canada). En raison des circonstance particulières, un exposé conjoint de Christine Meunier (Laboratoire Parole et Langage LPL, CNRS, Aix-en-Provence, France) et Christophe Stécoli (police technique et scientifique française) a dû être annulé et reporté à une journée spéciale en septembre 2020.
Ces actes regroupent les articles des conférences JEP (volume 1), TALN (volume 2), RÉCITAL (volume 3), les articles décrivant les démonstrations (volume 4), et les articles des ateliers DEFT (volume 5) et ÉTeRNAL (volume 6). Pour la première fois, un appel spécifique à résumés en français d'articles parus dans une sélection de conférences internationales en 2019 était également proposé (volume 4). Un appel spécifique apprenti·e·s chercheur·euse·s destiné aux étudiants de licence, de master, ou en première année de thèse a également été proposé, pour leur proposer des présentations courtes ou sous forme de poster de leurs projets.
Pour les JEP, 87 articles ont été soumis, parmi lesquels 74 ont été sélectionnés, soit un taux de sélection de 85%.
Pour TALN, 58 articles ont été soumis, parmi lesquels 37 ont été sélectionnés, soit un taux de sélection de 63%, dont 10 comme article longs (17% des soumissions) et 27 comme article courts dont 20 en présentation orale (34% des soumissions) et 7 en présentation poster (12% des soumissions).
Pour RÉCITAL, 22 articles ont été soumis, parmi lesquels 16 ont été sélectionnés, soit un taux de sélection de 73%.
Nous souhaitons vivement remercier toutes les personnes qui ont participé à ce travail de relecture et de sélection :

— l'ensemble des relecteurs (voir page xi),
— le comité de programme des JEP (voir page viii),
— le comité de programme de TALN (voir page ix),
— le comité de programme de RÉCITAL (voir page x).

Nous souhaitons également remercier nos sociétés savantes : l'AFCP, assurant la continuité des éditions successives des JEP, et l'ATALA, dont le CPerm (comité permanent) assure la continuité des éditions

successives de TALN.
Nous remercions le comité d'organisation et les nombreuses personnes qui ont assuré le soutien administratif et technique pour que cette conférence se déroule dans les meilleures conditions, et en particulier Yannick Parmentier pour son travail pour la diffusion de ces actes sur HAL et les différents sites d'archives ouvertes (anthologie ACL et talnarchives.atala.org/).
Nous remercions enfin tous les partenaires institutionnels et industriels qui nous ont fait confiance, en particulier l'université de Lorraine, le CNRS, l'Inria, le LORIA, l'ATILF, l'INIST, le master TAL de l'Institut des Sciences du Digital Management & Cognition (IDMC), le projet OLKI de l'initiative Lorraine Université d'Excellence (LUE), la Région Grand Est, *The Evaluations and Language resources Distribution Agency* (ELDA), le projet ANR PARSEME-FR, la délégation générale à la langue française et aux langues de France (DGLFLF), l'Association des Professionnels des Industries de la Langue (APIL) et les entreprises Synapse, Yseop et Orange.
Bonne conférence à toutes et à tous !

Les présidentes et présidents JEP : David Langlois et Slim Ouni
TALN : Chloé Braud et Sylvain Pogodalla
RÉCITAL : Christophe Benzitoun et Laurine Huber

Comités

Comité de programme des JEP

Martine Adda-Decker (Laboratoire de Phonétique et Phonologie, CNRS)
Jean-Francois Bonastre (LIA, Université d'Avignon)
Fethi Bougares (LIUM, Le Mans Université) Philippe Boula De Mareüil (LIMSI, CNRS)
Hervé Bredin (LIMSI, CNRS)
Olivier Crouzet (LLING, Université de Nantes)
Elisabeth Delais-Roussarie (LLING, Université de Nantes)
Véronique Delvaux (Laboratoire de Phonétique, IRSTL, Université de Mons)
Camille Fauth (LiLPa, Université de Strasbourg)
Emmanuel Ferragne (CLILLAC-ARP, Université de Paris)
Cecile Fougeron (Laboratoire de Phonétique et Phonologie, CNRS)
Corinne Fredouille (LIA, Université d'Avignon)
Alain Ghio (LPL, CNRS)
Camille Guinaudeau (LIMSI, Université Paris Sud)
Anne Guyot Talbot (CLILLAC-ARP, Université de Paris 7)
Bernard Harmegnies (Laboratoire de Phonétique, IRSTL, Université de Mons)
Nathalie Henrich Bernardoni (Gipsa-lab, CNRS)
Bassam Jabaian (LIA, Université d'Avignon)
David Langlois (LORIA, Université de Lorraine)
Yves Laprie (LORIA, CNRS)
Anthony Larcher (LIUM, Université du Maine)
Gwénolé Lecorvé (IRISA, Université de Rennes)
Benjamin Lecouteux (LIG, Université Grenoble Alpes)
Georges Linarès (LIA, Université d'Avignon)
Damien Lolive (IRISA, Université Rennes)
Julie Mauclair (IRIT)
Yohann Meynadier (LPL, Aix-Marseille Université)
Slim Ouni (LORIA, Université de Lorraine)
Thomas Pellegrini (IRIT, Université de Toulouse)
François Portet (LIG, Grenoble INP)
Fabian Santiago (Structures Formelles du Langage, Université de Paris 8)
Christophe Savariaux (Gipsa-lab, CNRS)
Nathalie Vallee (Gipsa-lab, Université Grenoble Alpes)
Ioana Vasilescu (LIMSI, CNRS)

Comités de programme TALN

Maxime Amblard (LORIA, Université de Lorraine)
Chloé Braud (IRIT, CNRS)
Caroline Brun (Naver Labs Europe)
Nathalie Camelin (LIUM, Université du Maine)
Marie Candito (Universtité Paris 7)
Vincent Claveau (IRISA, CNRS)
Chloé Clavel (Telecom-ParisTech)
Mathieu Constant (ATILF, CNRS, Université de Lorraine)
Pascal Denis (Inria)
Cécile Fabre (Université Toulouse 2)
Thomas François (Université catholique de Louvain)
Núria Gala (LPL, CNRS, Aix-Marseille Université)
Natalia Grabar (STL, CNRS, Université Lille 3)
Anne-Laure Ligozat (LIMSI, CNRS, ENSIIE, Université Paris-Saclay)
Emmanuel Morin (LINA, Université de Nantes)
Sylvain Pogodalla (LORIA, Inria)
Solen Quiniou (LINA, Université de Nantes)
Corentin Ribeyre (Etermind)
Tim van de Cruys (IRIT, CNRS)
Pierre Zweigenbaum (LIMSI, CNRS, Université Paris-Saclay)

Comité de programme RÉCITAL

Jean-Yves Antoine (Université François Rabelais de Tours)
Sonia Badene (Linagora, IRIT)
Frédéric Béchet (LIF, Aix Marseille Université)
Christophe Benzitoun (ATILF, Université de Lorraine)
Maria Boritchev (LORIA, Inria)
Léo Bouscarrat (EURA NOVA, Aix-Marseille Université)
Manon Cassier (INALCO, Paris)
Kevin Deturck (Viseo Technologies)
Emmanuelle Esperança-Rodier (GETALP, Université Grenoble Alpes)
Kim Gerdes (sorbonne nouvelle)
Nicolas Hernandez (LINA, UMR 6241, CNRS, Université de Nantes)
Lydia-Mai Ho-Dac (CLLE-ERSS, Université Toulouse Jean Jaurès)
Laurine Huber (LORIA, Université de Lorraine)
Sylvain Kahane (Modyco, Université Paris Ouest Nanterre)
Gwénolé Lecorvé (IRISA, Université de Rennes, CNRS)
Joël Legrand (LORIA, Inria, CNRS)
Anne-Laure Ligozat (LIMSI, CNRS, ENSIIE, Université Paris-Saclay)
Pierre Ludmann (LORIA, Université de Lorraine)
Yann Mathet (Université de Caen)
Anne-Lyse Minard (IRISA, CNRS)
Sandrine Ollinger (ATILF, UMR 7118, CNRS)
Yannick Parmentier (LORIA, Université de Lorraine)
Justine Reynaud (LORIA, Université de Lorraine)
Stella Zevio (LIPN, Université de Paris 13)

Relectrices et relecteurs

Gilles Adda (LIMSI, CNRS) Salah Ait-Mokhtar (Naver Labs Europe)
Charlotte Alazard (Université Toulouse 2 Jean Jaurès)
Alexandre Allauzen (LIMSI-CNRS, Université Paris-Sud)
Pascal Amsili (Université Paris Diderot)
Pierre André Hallé (Laboratoire de Phonétique et Phonologie, CNRS–Université Paris 3)
Régine André-Obrecht (Université Paul Sabatier Toulouse III)
Jean-Yves Antoine (Université François Rabelais de Tours)
Nicolas Audibert (Laboratoire de Phonétique et Phonologie, CNRS–Université Paris 3)
Nelly Barbot (IRISA, Université de Rennes 1)
Claude Barras (LIMSI, CNRS)
Loïc Barrault (University of Sheffield)
Katarina Bartkova (ATILF, Université de Lorraine)
Frédéric Béchet (LIF, Aix Marseille Université)
Nathalie Bedoin (DDL, Université Lyon 2)
Patrice Bellot (LSIS, CNRS, Aix-Marseille Université)
Asma Ben Abacha (National Library of Medicine, National Institutes of Health)
Delphine Bernhard (LiLPa, Université de Strasbourg)
Roxane Bertrand (LPL, CNRS, Aix-Marseille Université)
Laurent Besacier (Laboratoire d'Informatique de Grenoble)
Yves Bestgen (F.R.S-FNRS et Université Catholique de Louvain)
Frédéric Bimbot (IRISA, CNRS)
Caroline Bogliotti (MODYCO, UMR 7114, CNRS, Université Paris Nanterre)
Anne Bonneau (LORIA, CNRS)
Stéphanie Borel (Université de Tours)
Féthi Bougarès (LIUM, Le Mans Université)
Leila Boutora (Laboratoire Parole et Langage, Aix Marseille Université)
Paul Caillon (LORIA, Université de Lorraine)
Mélanie Canault (DDL, Université Lyon 2)
Thierry Charnois (LIPN, CNRS, Université de Paris 13)
Chloé Clavel (Telecom-ParisTech)
Maximin Coavoux (Université Grenoble Alpes, CNRS)
Vincent Colotte (LORIA, Université de Lorraine)
Juan Manuel Coria (LIMSI, Université Paris-Saclay Paris 13)
Benoît Crabbé (Université Paris 7)
Lise Crevier Buchman (Laboratoire de Phonétique et Phonologie, CNRS, Hôpital Foch)
Béatrice Daille (LINA, Université de Nantes)
Géraldine Damnati (Orange Labs)
Dan Dediu (Dynamique du Langage, UMR5596, Université Lumière Lyon 2)
Joseph Di Martino (LORIA, Université de Lorraine)
Gaël Dias (Université Caen Normandie)
Amazouz Djegdjiga (LPP, Université Sorbonne Nouvelle – Paris 3)
Benjamin Elie (IMSIA, ENSTA ParisTech)
Iris Eshkol-Taravella (Université d'Orléans)
Emmanuelle Esperança-Rodier (GETALP, Université Grenoble Alpes)
Yannick Estève (LIA, Université d'Avignon)
Dominique Estival (Western Sydney University)
Olivier Ferret (CEA LIST)
Lionel Fontan (Archean Labs)
Karën Fort (Sorbonne Université)
Claire Gardent (LORIA, CNRS)
Eric Gaussier (LIG, Université Grenoble Alpes)
Cédric Gendrot (LPP, Université Sorbonne Nouvelle – Paris 3)
James German (Laboratoire Parole et Langage, Aix Marseille Université)
Cyril Goutte (National Research Council Canada)
Cyril Grouin (LIMSI, CNRS, Université Paris-Saclay)
Pierre André Hallé (LPP, Université Sorbonne Nouvelle – Paris 3)
Olivier Hamon (Syllabs)
Thierry Hamon (LIMSI, Université Paris-Saclay, CNRS, Université Sorbonne Paris Nord)
Bernard Harmegnies (Institut de Recherche en Sciences et Technologies du Langage, Université de Mons)
Nabil Hathout (CLLE, CNRS)
Amir Hazem (LS2N, Université de Nantes)
Nicolas Hernandez (LS2N, Université de Nantes)
Fabrice Hirsch (Praxiling, Université Paul Valéry Montpellier 3)
Thomas Hueber (GIPSA-lab, CNRS)
Kathy Huet (Institut de Recherche en Sciences et Technologies du Langage, Université de Mons)

Stéphane Huet (LIA, Université d'Avignon)
Mathilde Hutin (LIMSI, Université Paris-Saclay, CNRS, Université Sorbonne Paris Nord)
Irina Illina (LORIA, Université de Lorraine)
Christine Jacquin (LS2N Université de Nantes)
Adèle Jatteau (STL, UMR 8163, Université de Lille, CNRS)
Denis Jouvet (LORIA, Inria)
Sylvain Kahane (Modyco, Université Paris Ouest Nanterre)
Takeki Kamiyama (LPP, Université Paris 8 Vincennes-Saint-Denis)
Hannah King (CLILLAC-ARP, Université Paris Diderot)
Olivier Kraif (Université Grenoble Alpes)
Matthieu Labeau (Telecom Paris)
Mathieu Lafourcade (LIRMM, Université de Montpellier)
Mohamed Lahrouchi (SFL, UMR 7023, CNRS Université Paris 8)
Muriel Lalain (LPL, CNRS, Aix-Marseille Université)
Joseph Lark (Dictanova)
Thomas Lavergne (LIMSI, CNRS, Univ. Paris Sud, Université Paris Saclay)
Guillaume Le Berre (LORIA, Université de Lorraine)
Gwénolé Lecorvé (IRISA, Université de Rennes, CNRS)
Benjamin Lecouteux (Laboratoire Informatique de Grenoble)
Claire Lemaire (Université Grenoble Alpes)
Yves Lepage (Waseda University)
Joseph Le Roux (LIPN, Université de Paris 13)
Veronika Lux (ATILF, CNRS)
Paolo Mairano (STL, UMR 8163, Université de Lille)
Anna Marczyk (LPL, CNRS, Aix-Marseille Université)
Denis Maurel (Université François Rabelais de Tours)
Christine Meunier (LPL, CNRS, Aix-Marseille Université)
Alexis Michaud (LACITO, CNRS)
Richard Moot (LIRMM, CNRS)
Véronique Moriceau (LIMSI, CNRS)
Philippe Muller (IRIT, Université de Toulouse)
Alexis Nasr (LIF, Université de la Méditerranée)
Sylvain Navarro (CLLE-ERSS, CNRS)
Luka Nerima (Université de Genève)
Aurélie Névéol (LIMSI, CNRS, Université Paris-Saclay)
Jian-Yun Nie (Université de Montreal)
Damien Nouvel (INaLCO)
Nicolas Obin (IRCAM)
Yannick Parmentier (LORIA, Université de Lorraine)
Sebastian Peña Saldarriaga (Dictanova)
Marie Philippart de Foy (Université de Mons)
Myriam Piccaluga (Institut de Recherche en Sciences et Technologies du Langage, Université de Mons)
Claire Pillot-Loiseau (LPP, UMR 7018, CNRS, Université Sorbonne Nouvelle – Paris 3)
Serge Pinto (LPL, CNRS, Aix-Marseille Université)
Agnès Piquard (LORIA, CNRS, Université de Lorraine)
Thierry Poibeau (LaTTiCe, CNRS)
Alain Polguère (ATILF Université de Lorraine)
Laurent Prévot (LPL, CNRS, Aix-Marseille Université)
Jean-Philippe Prost (LIRMM, Université de Montpellier)
Christian Raymond (IRISA, INSA de Rennes)
Christian Retoré (LIRMM, Université de Montpellier)
Albert Rilliard (LIMSI, CNRS, Université Paris-Saclay)
Virginie Roland (Institut de Recherche en Sciences et Technologies du Langage, Université de Mons)
Sophie Rosset (LIMSI, CNRS, Université Paris-Saclay)
Véronique Sabadell (LPC, Aix Marseille Université)
Stéphane Schneider (INIST, CNRS)
Didier Schwab (Université Grenoble Alpes)
Pascale Sébillot (IRISA, INSA de Rennes)
Djamé Seddah (Almanach, Université Paris la Sorbonne)
Gilles Serasset (LIG, Université Grenoble Alpes)
Romain Serizel (LORIA, Université de Lorraine)
Kamel Smaïli (LORIA, Université de Lorraine)
Rudolph Sock (LiLPa, Université de Strasbourg)
Ludovic Tanguy (CLLE, CNRS)
Xavier Tannier (LIMICS, Sorbonne Université, INSERM)
Andon Tchechmedjiev (IMR, Mines Alès)
Juan-Manuel Torres-Moreno (LIA, Université d'Avignon)
Nicolas Turenne (LISIS, INRA)
Béatrice Vaxelaire (LiLPa, Université de Strasbourg)

Anne Vilain (GIPSA-lab, Université de Grenoble Alpes)
Coriandre Vilain (GIPSA-lab, Université de Grenoble Alpes)
Guillaume Wisniewski (LLF, Université de Paris)
Jane Wottawa (LIUM, Le Mans Université)
Yaru Wu (LPP, MoDyCo, Université Paris Nanterre)
Kossi Seto Yibokou (LiLPa, Université de Strasbourg)
François Yvon (LIMSI, CNRS, Université Paris-Sud)

Table des matières

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Segmentation de texte non-supervisée pour la détection de thématiques à l'aide de plongements lexicaux

Alexandra Benamar[1, 2]
(1) Université Paris-Saclay, CNRS, LIMSI, 91405 Orsay
(2) EDF R&D, 7 Boulevard Gaspard Monge, 91120 Palaiseau
alexandra.benamar@{limsi.fr,edf.fr}

RÉSUMÉ

Cet article présente les principales méthodes de segmentation automatique de documents textuels spécifiques. La tâche de segmentation thématique de texte consiste à analyser un document pour en extraire des sections cohérentes. Les méthodes de segmentation non supervisées cherchent à optimiser une fonction de probabilité de segmentation ou une fonction de similarité qui peut être calculée entre les blocs ou au sein des blocs. Elles sont réparties en trois catégories : les méthodes statistiques, les méthodes à base de graphes et les approches neuronales. Parmi les approches neuronales utilisées, nous nous intéressons tout particulièrement à celles qui utilisent des plongements lexicaux pour représenter des phrases et définir des segments thématiques. Tout d'abord, nous montrons que les plongements lexicaux permettent une amélioration nette des performances par rapport à des méthodes statistiques. Ensuite, nous évaluons l'impact du choix de la représentation vectorielle des phrases pour cette tâche de segmentation non supervisée.

ABSTRACT

Unsupervised text segmentation for topic detection using embeddings

This paper presents the state of the art of automatic segmentation of domain-specific documents. The task of topic segmentation consists in analyzing a document in order to extract coherent sections. The aim of unsupervised segmentation methods is either to maximize a probabilistic function or to use a similarity function that can be maximize between or within segmented blocks. They are divided into three categories : statistical methods, graph-based methods and neural approaches. Among the last ones, we are particularly interested in those which use latent representations of words to define segments. First, we show that embeddings allow a significant improvement in performance compared to statistical methods. Next, we assess the impact of sentence representation on unsupervised text segmentation methods.

MOTS-CLÉS : segmentation de texte ; méthodes non-supervisées ; plongements lexicaux.

KEYWORDS: text segmentation ; unsupervised methods ; embeddings.

1 Introduction

La segmentation de texte en thématiques cohérentes est une tâche fondamentale en traitement automatique des langues (TAL), à différentes granularités. Un ensemble d'approches appelées segmentation thématique (*topic segmentation* en anglais) s'emploie à diviser un document en segments thématiquement cohérents. Ces approches sont souvent considérées comme un pré-requis pour d'autres

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 1–14.
HAL : hal-02786182.

tâches telles que l'analyse du discours (Polanyi *et al.*, 2004) et l'analyse de sentiments (Mu *et al.*, 2012). La segmentation en thèmes permet de répondre à des problématiques variées comme le résumé automatique (Chuang & Yang, 2000; Angheluta *et al.*, 2002), la recherche d'information (Huang *et al.*, 2003; Prince & Labadié, 2007) et l'analyse de dialogues (Song *et al.*, 2016). À un niveau plus fin, la segmentation de texte consiste à segmenter des phrases en unités élémentaires du discours (EDU pour *Elementary Discourse Units*) (Marcu, 2000) définies comme des unités similaires à des clauses servant de blocs élémentaires pour la segmentation du discours.

Afin de répondre à ces problématiques de segmentation de texte, des méthodes supervisées et non supervisées de fouille de texte sont généralement mises en œuvre. Les méthodes supervisées nécessitent un grand volume de données d'apprentissage spécifiques à un domaine d'application très structurées (documents juridiques, rapports médicaux, ...) et sont sensibles au bruit. Dans cet article, nous nous intéressons à la segmentation de texte non supervisée, où la spécificité de la méthode réside dans le choix de la fonction objectif et dans la représentation des phrases du corpus (dense ou latente). Depuis de nombreuses années, différentes approches ont été explorées pour segmenter des textes comme la détection de cooccurences de mots entre des segments, la détection de thématiques, l'utilisation de représentations de phrases ou de mots sous forme de graphes et l'utilisation d'architectures neuronales. Ici, nous nous intéressons à l'utilisation de plongements lexicaux pour la segmentation automatique de texte en sections thématiques pertinentes dans des corpus de données spécifiques. Les plongements lexicaux ont connu un grand succès ces dernières années sur plusieurs tâches d'apprentissage non supervisé, en permettant une représentation syntaxique et sémantique des mots dans un corpus. Les évaluations seront réalisées sur le corpus construit lors de la compétition Défi fouilles de texte (DEFT) de l'édition 2006 (Heitz *et al.*, 2006) composé de discours politiques et de textes juridiques. Les méthodes testées dans cet article ont été retenues pour cette tâche spécifique à partir d'un état de l'art assez large des méthodes neuronales de segmentation de texte et comparées à des méthodes statistiques fréquemment utilisées. Après avoir présenté un état de l'art des méthodes de segmentation thématique de documents et de représentations du langage (section 2), nous décrivons le corpus DEFT utilisé (section 3), puis nous présentons les expériences réalisées pour cette tâche (section 4) avant de conclure et de définir nos perspectives de recherche (section 5).

2 État de l'art

Dans cette section, nous présentons tout d'abord les grandes familles de méthodes non supervisées de segmentation de texte. Ensuite, nous nous intéressons aux évolutions récentes qui permettent d'améliorer les représentations de texte : les plongements lexicaux. Puis, nous détaillons deux méthodes proposées ces dernières années pour utiliser des plongements de mots pour la segmentation automatique de texte. Enfin, nous nous intéressons aux méthodes courantes d'évaluation des modèles de segmentation.

2.1 Méthodes de segmentation non supervisée

Parmi les approches répandues de segmentation linéaire de texte, les méthodes non supervisées constituent un champ de recherche majoritaire. Ces méthodes englobent la cohésion lexicale, la modélisation statistique, des méthodes par propagation d'affinité et des approches de *topic modeling*.

2.1.1 Objectif des méthodes

On distingue deux stratégies d'optimisation de la segmentation de texte : des méthodes probabilistes et des approches basées sur la similarité de contenu. La première approche définit un modèle de langue à travers l'étude de la distribution de probabilités des mots d'un corpus en vue d'affecter chaque mot à une thématique. En 2001, une approche probabiliste (Utiyama & Isahara, 2001) consiste à maximiser la probabilité de segmentation d'un texte. Formellement, si on considère une séquence de mots $W = w_1 w_2 \dots w_n$ et une segmentation $S = s_1 s_2 \dots s_m$ de W, l'objectif est de maximiser :

$$P(S|W) = \frac{P(S)P(W|S)}{P(W)} \tag{1}$$

Cela revient à trouver la séquence de segments $S = argmax_s P(W|S)P(S)$, $P(W)$ étant un terme constant. Afin d'évaluer cet objectif, les auteurs émettent deux hypothèses :

1. Les segments sont statistiquement indépendants les uns des autres.
2. Étant donné un segment, les mots au sein d'un segment sont statistiquement indépendants les uns des autres.

Cette méthode est la première a avoir été proposée pour traiter la segmentation comme un problème d'optimisation probabiliste. Elle a permis d'obtenir des résultats à l'état de l'art dans des applications de résumés automatiques. Cependant, cette approche ne semble pas adaptée à une application de recherche d'information. En effet, l'algorithme découpe les gros documents en peu de segments, ce qui est intéressant pour le résumé automatique (où on veut détecter le moins de thématiques pertinentes possibles), mais ce qui n'est pas pertinent pour l'extraction de petits segments de texte.

La deuxième approche consiste à étudier la similarité lexicale entre différentes unités de texte. Les unités de texte peuvent être des mots, des phrases, des séquences de mots ou des paragraphes. Le plus souvent, la similarité est mesurée entre des blocs de textes, sous forme de similarité cosinus $S(s_i, s_j)$ entre deux segments $s_i = w_1 w_2 \dots w_n$:

$$S(s_i, s_j) = \frac{s_i \cdot s_j}{\|s_i\| \|s_j\|} \tag{2}$$

Dans l'équation, $s_i \cdot s_j$ est le produit vectoriel des deux vecteurs et $\|s_i\|$ est la norme L2 du vecteur s_i. La plupart des algorithmes de segmentation supposent que des fragments de texte ayant des distributions de mots similaires appartiendront à la même thématique. Dans les approches de similarité, on distingue des blocs similaires et homogènes en utilisant la similarité des éléments au sein d'un même bloc. Deux approches sont étudiées pour évaluer l'homogénéité des blocs : la maximisation de la similarité au sein des blocs (Reynar, 1998; Choi, 2000) et la maximisation de la dissimilarité entre des blocs (Reynar, 1998). Idéalement, les deux stratégies seraient combinées pour trouver des frontières de segmentations optimales.

2.1.2 Familles de méthodes

Les grandes familles de méthodes de segmentation de texte non supervisées sont :

1. Les méthodes statistiques : elles formulent l'hypothèse que des segments différents contiennent des mots différents. Elles caractérisent les mots par des méthodes statistiques (*ie.* TF, IDF, ...).

2. Les modèles à base de graphes : les nœuds représentent les phrases, et les arêtes peuvent représenter le nombre de mots qu'elles partagent (Pourvali & Abadeh, 2012) ou encore des liens de similarité vectorielle entre leurs mots communs (Glavaš *et al.*, 2016).
3. Les méthodes basées sur des réseaux de neurones.

Dans la suite de cet article, nous détaillerons les méthodes statistiques, servant de base à notre étude, et les méthodes neuronales au cœur de celle-ci.

Modèles statistiques Un document est souvent constitué de sections cohérentes, contenant elles-mêmes plusieurs unités textuelles (paragraphes, phrases, segments, etc.) (Salton *et al.*, 1996). Une hypothèse sous-jacente à la plupart des algorithmes de segmentation de documents consiste à dire que les répétitions lexicales signifient la continuité d'une thématique, alors que le changement de distribution lexicale indique une transition vers une autre thématique. Ce principe a été formalisé par Halliday & Hasan (1976) dans la théorie de la cohésion. Généralement, si deux thématiques sont suffisamment différentes, le vocabulaire significatif associé à chacun d'entre-eux sera également différent. De plus, si des mots apparaissent dans des contextes similaires, ils seront probablement liés sémantiquement, ce qui permet l'utilisation de modèles basés sur la cooccurrences de mots. Reynar (1998) observe que les anaphores pronominales ont tendance à survenir plus fréquemment au sein de segments qu'entre différents segments et montre ainsi que la segmentation en thématiques pourrait être une étape cruciale pour la résolution d'anaphores pronominales. Les premières méthodes de segmentation de textes consistaient à représenter la cohésion lexicale dans un espace vectoriel, en explorant des modèles basés sur la cooccurences de mots. Parmi elles, TextTiling (Hearst, 1997) est une technique de segmentation de texte basée sur la similitude entre des blocs de mots adjacents, en se basant uniquement sur la cooccurence des mots et leur distribution. Cette approche sous-entend qu'une baisse de la similarité entre blocs adjacents correspond à un changement de thématique. L'inconvénient majeur de cette méthode est qu'elle considère la similarité entre des blocs adjacents, sans considérer des dépendances à long terme. Choi (2000) améliore TextTiling avec l'algorithme C99 et calcule un score de cohérence entre toutes les paires d'unités (phrases ou segments) d'un texte au lieu de seulement étudier la cohérence entre les unités voisines. L'algorithme recherche les frontières qui optimisent une fonction objectif basée sur le score de cohérence, en effectuant des coupures successives du texte, similaire à du *clustering* hiérarchique descendant. Plus récemment, des méthodes de segmentation basées sur des modèles de *topic modeling* ont vu le jour. Parmi elles, la méthode PLDA (Purver *et al.*, 2006) calcule des frontières de segmentation et un modèle d'allocation de Dirichlet latente.

Réseaux de neurones Les réseaux de neurones profonds (*Deep Neural Networks*) ont connu un grand succès en TAL, notamment avec l'utilisation de méthodes de *transfer learning* et de modèles de langues. Ces architectures ont permis des gains de performances impressionnants, notamment en classification supervisée, en reconnaissance d'entités nommées ou encore en *Question Answering*. Depuis, les réseaux de neurones ont été introduits pour des tâches de segmentation de données textuelles. Par exemple, les réseaux de neurones récurrents LSTM ont été utilisés pour la segmentation de textes (Koshorek *et al.*, 2018) en considérant ce problème comme une tâche d'apprentissage supervisé, où chaque phrase de fin de segment est étiquetée par un marqueur spécifique. Les auteurs ont utilisé un premier LSTM bidirectionnel pour construire des représentations de phrases à partir de mots et un deuxième LSTM pour attribuer une probabilité d'appartenance de chaque phrase aux labels. Au vu du récent succès des plongements lexicaux, un champ de recherche s'intéresse à leur incorporation à la tâche de segmentation automatique de textes.

2.2 Représentation du langage

Les représentations vectorielles de mots, qui ont permis des gains de performance importants sur plusieurs tâches comme la classification, consistent à associer un mot à un vecteur de taille fixe. Parmi les modèles les plus connus figurent Word2Vec (Mikolov *et al.*, 2013), fondé sur un modèle de prédiction d'un mot à partir du contexte de mots environnants, et GloVe (Pennington *et al.*, 2014) basé sur la prédiction des cooccurences de mots dans un document. Ces approches permettent d'appréhender des représentations figées des mots à partir de leur contexte environnant. Toutefois, les résultats obtenus ne rendent pas compte de la richesse syntaxique et sémantique de la langue. Les mots polysémiques seront représentés par un seul vecteur, indépendamment de leur sens réel dans la phrase. De plus, les mots n'appartenant pas au vocabulaire de l'ensemble d'apprentissage (*Out Of Vocabulary* en anglais ou OOV) ne seront pas reconnus par le modèle ce qui oblige l'utilisateur à construire de nouveaux modèles pour le traitement de domaines spécifiques.

Plusieurs méthodes tirent parti des réseaux de neurones récurrents pour construire des représentations plus riches basées sur des modèles de langues statistiques. Ceux-ci consistent à calculer à partir d'un enchaînement de mots la probabilité d'apparition du mot suivant. ELMo (*Embeddings from Language Models*) (Peters *et al.*, 2018) construit des plongements de mots contextuels qui utilisent la richesse syntaxique et sémantique des mots. Ce modèle combine des réseaux de neurones récurrents pour l'analyse des mots et des réseaux de neurones convolutifs pour l'analyse des chaînes de caractères. Chaque caractère est ainsi représenté par un vecteur de petite taille et chaque mot est représenté par la concaténation des vecteurs des caractères le composant. Cela permet de représenter tous les mots qui n'ont jamais été vus par le système (OOV). De plus, l'utilisation de réseaux récurrents bidirectionnels permet l'ajout d'informations syntaxiques et sémantiques à travers l'enrichissement du contexte pour chaque mot. En utilisant des méthodes de combinaison linéaire et de concaténation vectorielle des couches cachées pour obtenir de nouvelles caractéristiques, ELMo permet une amélioration significative des résultats sur des tâches de *Question Answering* (QA), de détection d'informations sémantiques « qui a fait quoi à qui ? », de résolution de coréférence, etc.

Méthode sémantique Alemi & Ginsparg (2015) démontrent l'utilité des plongements lexicaux sémantiques pour la segmentation de texte, à la fois dans les algorithmes existants et dans de nouveaux algorithmes de segmentation. Les auteurs ont incorporé à l'algorithme C99 (présenté en section 2.1) des plongements de mots GloVe. Cela a permis d'améliorer de quelques pourcents les résultats obtenus avec l'algorithme natif. Le corpus est tout d'abord nettoyé en supprimant la ponctuation et les mots vides. Puis, chaque mot est représenté par son vecteur GloVe associé et pondéré par l'Inverse Document Frequency (IDF) du mot dans l'ensemble du corpus. Enfin, chaque phrase est représentée par le vecteur moyen de l'ensemble des mots la composant. La segmentation est une forme de *clustering* donc un choix naturel pour la fonction de score est la somme des écarts carrés par rapport à la moyenne du segment, comme utilisée dans l'algorithme k-means. En général, la similitude cosinus est utilisée pour le vecteur de mots, mais les auteurs ont choisi de normaliser les vecteurs afin d'utiliser la métrique euclidienne, ce qui se rapproche plus de l'algorithme k-means par défaut. Enfin, les auteurs présentent différentes stratégies d'optimisation de la fonction objectif : une méthode d'optimisation dite « gourmande »(*greedy* en anglais) et une méthode d'optimisation dynamique. La méthode gourmande consiste à diviser le texte itérativement à l'endroit où le gain est le plus grand, jusqu'à ce que ce gain soit inférieur à un seuil de pénalité donné. Le gain est défini comme la somme des normes des segments gauche et droit moins la norme du segment à diviser. La méthode gourmande consiste à construire de manière itérative une structure de données qui stocke les

résultat d'un fractionnement optimal. Il en résulte une matrice stockant un score pour un segment de la position i à j, étant donné une segmentation optimale jusqu'à i. La méthode dynamique a permis une amélioration des résultats de l'algorithme C99 couplé à des plongements lexicaux.

Plus récemment, les architectures neuronales *transformer* (Vaswani *et al.*, 2017) se sont imposées comme une alternative performante aux réseaux de neurones récurrents (Cho *et al.*, 2014), grâce à l'efficacité de leur apprentissage et à leurs performances supérieures en termes de capture des dépendances longue distance. Les modèles *Universal Sentence Encoder* (USE) sont des modèles pré-entrainés d'*embeddings* à différents niveaux : mots, phrases et paragraphes et ont été testés sur des tâches supervisées telles que la classification de sentiments ainsi que sur des tâches de similarité entre les documents (Cer *et al.*, 2018). Parmi les nombreux modèles proposés par USE, le plus performant est le modèle transformer qui utilise les mécanismes d'attention pour calculer des représentations vectorielles sensibles au contexte des mots. En 2018, Devlin *et al.* (2018) proposent BERT, un modèle transformer bi-directionnel à l'état de l'art sur 11 tâches de TAL, dont le QA où il dépasse même l'annotation humaine. BERT contient deux nouvelles tâches de pré-entraînement : l'une à l'échelle des mots et l'autre à l'échelle des phrases.

2.3 Plongements lexicaux pour la segmentation de texte

Méthode séquentielle D'autres travaux (Karus, 2019) présentent un modèle de segmentation séquentiel qui utilise une fenêtre glissante sur les phrases d'un corpus pour détecter les frontières de segmentation (Figure 1). Tout d'abord, l'auteur représente chaque phrase du corpus par un plongement de phrase, calculé avec Word2Vec, en calculant la moyenne des vecteurs de mots la composant. Ensuite, l'algorithme observe les plongements des n premières phrases (n correspond à la taille de la fenêtre) et calcule la plus grande distance cosinus entre les segments de phrases puis divise le texte au niveau de cet indice. Puis, il traite ensuite les n phrases après la scission décidée et répète le processus jusqu'à ce qu'il atteigne la fin du texte. L'auteur a également essayé de réduire la dimension des algorithmes avec une analyse en composantes principales, ce qui lui a permis de gagner en performance.

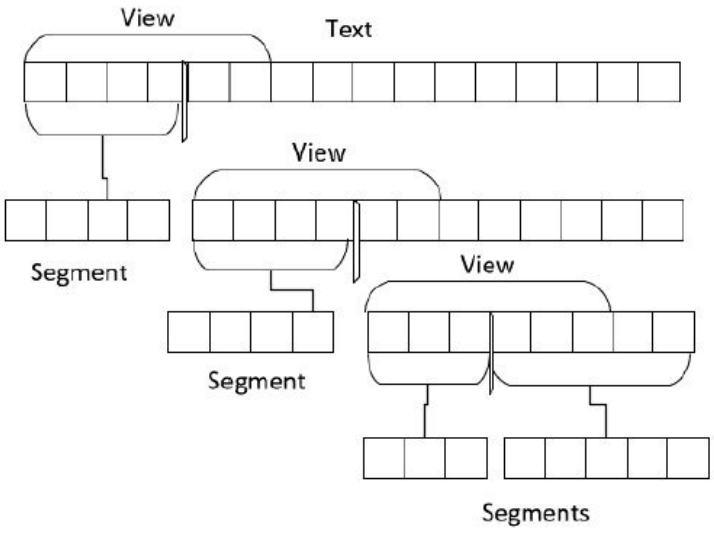


FIGURE 1 – Segmentation séquentielle de textes

2.4 Méthodes d'évaluation

L'évaluation en segmentation de texte est une tâche difficile qui peut prendre en compte différents paramètres. Parmi eux, certains permettent de détecter des erreurs de labels (précision, rappel et f-mesure), d'autres calculent des erreurs de labels et de frontières (P_k, WindowDiff, ...) et les dernières calculent des erreurs de frontières et de types (SER). En segmentation automatique, des étiquettes 0-1 sont associées aux phrases du corpus. L'étiquette 1 signifie que la phrase est la première d'un segment et l'étiquette 0 signifie que la phrase se trouve au sein d'un segment.

Habituellement utilisées en recherche d'information (Van Rijsbergen, 1979), les méthodes de précision et de rappel ont été appliquées à l'évaluation d'algorithmes de segmentation de textes. Le rappel est défini comme le pourcentage de frontières correctement détectées par l'algorithme et la précision est le pourcentage de frontières identifiées par l'algorithme qui correspondent à des coupures réelles du corpus. Formellement :

$$\text{Précision} = \frac{\textit{Nombre de segments correctement étiquetés}}{\textit{Nombre de segments fournis}} \tag{3}$$

$$\text{Rappel} = \frac{\textit{Nombre de segments correctement étiquetés}}{\textit{Nombre de segments étiquetés dans la référence}} \tag{4}$$

Néanmoins, la relation entre la précision et le rappel tendent à rendre cette tâche difficile. En effet, l'ajout de frontières tendra à améliorer le rappel et à diminuer la précision. En plus du rappel et de la précision, la f-mesure a également été utilisée par Baeza-Yates *et al.* (1999) afin de combiner la précision et le rappel mais l'inconvénient de cette approche est que les résultats sont difficiles à interpréter (Baeza-Yates *et al.*, 1999). De plus, le rappel et la précision ne tiennent pas compte des pénalités liées à la distance à la segmentation réelle obtenue par les algorithmes. Par exemple, soit une segmentation réelle $S = 10101010$ et deux algorithmes de segmentation $Seg_1 = 11000110$ et $Seg_2 = 11110000$, on devrait considérer que l'algorithme Seg_1 est meilleur que Seg_2 parce qu'il construit des séquences plus similaires aux segments réels que Seg_2, ce que ces méthodes ne permettent pas de déterminer. De plus, les segments produits par Seg_1 sont plus similaires à S en termes de taille que ceux produits par Seg_2.

En 1997, une nouvelle méthode d'évaluation P_k est proposée par Beeferman *et al.* (1997) afin de pallier ces problèmes et inclure un système de pénalités plus juste. Elle est calculée en fixant k comme étant la moitié de la moyenne de la taille des segments réels, et en déplaçant une fenêtre de taille k sur le jeu de données. A chaque endroit, l'algorithme détermine si les segments en début et fin de fenêtre appartiennent au même segment ou non, et incrémente un compteur si la réponse est fausse. Le nombre résultant est mis à l'échelle entre 0 et 1 en divisant par le nombre de mesures prises. Un algorithme qui attribue correctement toutes les frontières reçoit un score de 0. Pour justifier cette méthode, Beeferman *et al.* (1999) insistent sur leur volonté d'empêcher les algorithmes de s'adapter aux méthodes d'évaluation. Pour cela, les algorithmes dégénérés qui placent des frontières à chaque position ou ne placent aucune frontière obtiennent approximativement le même score. De plus, les auteurs définissent un faux négatif (également appelé un « raté ») comme un cas où une frontière est présente dans la segmentation de référence mais manquante dans la segmentation hypothétique de l'algorithme, et une assignation faussement positive d'une frontière qui n'existe pas dans la segmentation de référence. L'interprétation de P_k est la suivante : plus il est petit, plus le résultat obtenu se rapproche du résultat de référence.

Plus récemment, Pevzner & Hearst (2002) ont mis en avant des problèmes liés à la méthode P_k : les faux-négatifs sont plus pénalisés que les faux-positifs, le nombre de frontières n'est pas considéré, la méthode est difficilement interprétable lorsqu'elle est supérieure à 0, etc. Ils proposent alors *WindowDiff*, inspiré de P_k et modifient la mesure d'erreur de l'algorithme tout en conservant la caractéristique souhaitable de pénaliser les presque-accidents moins que les faux positifs-purs et les faux-négatifs purs. Le correctif fonctionne comme suit : pour chaque position de la fenêtre, il suffit de comparer le nombre de frontières de segmentation de référence tombées dans cet intervalle (r_i) par rapport au nombre de frontières attribuées par l'algorithme (a_i). L'algorithme est pénalisé si $r_i = a_i$ (qui est calculé comme $|r_i - a_i| > 0$). Formellement :

$$\textit{WindowDiff}(ref, hyp) = \frac{1}{N-k} \sum_{i=1}^{N-k} (|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0) \tag{5}$$

où $b(i, j)$ représente le nombre de frontières entre les positions i et j dans le texte et N représente les fausses pénalités négatives observées dans la méthode P_k. Il capture également les faux positifs et les faux négatifs dans des segments de longueur inférieure à k. Comme pour P_k, WindowDiff est une erreur, donc plus sa valeur est petite, meilleur est le résultat.

Enfin, le *Slot Error Rate* (SER) (Makhoul *et al.*, 1999), fréquemment utilisé en détection d'entités nommées, combine et pondère différents types d'erreurs : insertions (I), délétions (D), erreurs de frontière (F), erreurs de type (T) et erreurs de type et de frontière (TF) et est calculé comme ceci :

$$\textit{SER} = \frac{I + D + TF + 0,5 * (T + F)}{\textit{Nombre de segments étiquetés de la référence}} \tag{6}$$

Cette méthode permet une évaluation rigoureuse de blocs de segments. Cependant, la particularité de la segmentation est qu'il ne s'agit pas de faire de distinction de labels entre deux segments A et B, toutes les phrases d'un segment sont représentées par des 0 et les frontières sont représentées par des 1. En conclusion, cette méthode, bien que complète et interprétable, ne semble pas pertinente pour une tâche de segmentation de texte.

3 Corpus

Le corpus de données utilisé dans cet article a été construit dans le cadre de l'édition DEFT 2006 (Heitz *et al.*, 2006) pour effectuer de la reconnaissance automatique de segments thématiques dans des textes rédigés en français. Nous avons choisi d'utiliser des corpus de domaines spécifiques pour effectuer de la segmentation thématique ce qui nous a conduit à sélectionner les genres suivants : discours politiques et textes juridiques.

Le premier corpus est composé de transcriptions manuelles de discours politiques prononcés par différents Présidents de la République française (Valéry Giscard d'Estaing, François Mitterrand et Jacques Chirac). Les en-têtes des discours comprenant le titre, la date et l'orateur ont été supprimés. Les auteurs ont capitalisé sur les discours de Valéry Giscard d'Estaing, quasiment tous présents dans le corpus, contrairement aux autres Présidents. Cet ensemble de données contient également des entretiens politiques avec des journalistes ou d'autres hommes politiques, tant que l'un des interlocuteurs est l'un des Présidents listés précédemment. La spécificité de ce jeu de données est

qu'il est entièrement en majuscules. La segmentation de référence consiste dans les paragraphes des textes originaux (volumétrie : environ 70 Mo).

Le deuxième corpus est composé d'articles de lois de l'Union Européenne. Les références aux images, les en-têtes, l'article final de signature et les lois de moins de 10 phrases ont été supprimés par les auteurs. Les références sont écrites sous la forme [REFERENCE], comme par exemple [EMPLACEMENT TABLE]. Les numéros des articles, chapitres, titres et annexes ont été remplacés par la lettre « X », comme par exemple « Article X ». Les segments thématiques de référence sont les lois qui peuvent être composées de plusieurs articles (volumétrie : environ 110 Mo).

La Table 1 présente des statistiques descriptives du corpus effectuées par les auteurs (Hurault-Plantet *et al.*, 2006). Les corpus sont très différents en terme de contenu des segments thématiques à retrouver. Les discours politiques contiennent beaucoup de segments courts alors que les textes juridiques contiennent deux fois moins de segments de plus grande taille (1). Malgré une volumétrie nettement plus grande pour les textes juridiques en termes de nombre total de phrases (1) et de mots (1), on note que le vocabulaire est moins riche que dans les discours politiques (1), ce qui signifie que la richesse lexicale est faible dans les articles de lois.

	#phrases	**#segs**	**#phrases/seg.**	**vocab.**	**#mots**	**#mots/phrase**	**#mots/seg.**
Discours	303 373	18 929	16	62 465	8 186 044	27	432
Lois	433 456	9 934	44	57 763	11 555 852	27	1 163

TABLE 1 – Statistiques descriptives sur les mots du corpus d'apprentissage. seg(s). : segment(s) ; vocab. : vocabulaire.

4 Expériences

Afin d'évaluer l'impact des plongements lexicaux sur une tâche de segmentation de texte, nous implémentons les méthodes sémantiques et séquentielles présentées en section (2.3). Cette approche sera comparée à des méthodes statistiques usuelles avec les algorithmes TextTiling et C99 présentés précédemment. Le deuxième objectif de l'article est de comparer des représentations vectorielles pour la segmentation. Pour cela, nous utiliserons la méthode séquentielle et nous comparerons la représentation non contextuelle Word2Vec (utilisée dans le papier d'origine) avec les représentations contextuelles ELMo et Universal Sentence Encoder.

4.1 Pré-traitements

Pour toutes les méthodes, la ponctuation a été supprimée. Les méthodes statistiques testées dans cet article (TextTiling et C99) sont appliquées dans la littérature après une phase de *stemming* de données. En français, cette méthode n'est pas toujours pertinente. Pour cette raison, nous avons remplacé l'étape de *stemming* par une lemmatisation du corpus avec l'outil TreeTagger [1]. De plus, ces documents sont rédigés en majuscules. Dans cet article, nous avons fait le choix de transformer les textes en minuscules. L'étape de lemmatisation a été réalisée avec l'outil sur les textes après transformation en minuscules de ceux-ci.

1. https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

4.2 Vectorisation

Les modèles vectoriels utilisés dans cet article sont Word2Vec, ELMo et Universal Sentence Encoder. Le modèle Word2vec choisi a été réalisé par Fauconnier (2015) et a été appris sur une sortie Wikipédia contenant 600 millions de mots. Il s'agit d'un modèle skip-gram contenant des vecteurs de taille 1000 et un vocabulaire de mots apparaissant au moins 200 fois dans le corpus. Le modèle ELMo utilisé (Che *et al.*, 2018) a été pré-entraîné sur 20 millions de mots en français sur un corpus Wikipédia et sur le corpus Common Crawl.

4.3 Paramètres

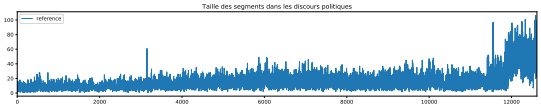
Les méthodes de segmentation ont été lancées sur l'ensemble de données d'apprentissage afin de trouver les paramètres permettant d'optimiser les résultats obtenus. Ces paramètres de calcul ont ensuite été appliqués au jeu de données de test pour l'évaluation des performances des méthodes. Les méthodes TextTiling et C99 ont été utilisées avec des fenêtres de calcul des frontières de taille 10 pour les discours politiques et 35 pour les textes juridiques. La taille optimale des segments utilisée pour la méthode de plongements sémantiques a été fixée à 16 pour les discours politiques et 40 pour les textes juridiques. Enfin, pour la méthode séquentielle, la fenêtre glissante est de taille 35 pour les discours politiques et elle est de taille 20 pour les textes juridiques. La réduction de dimension a été réalisée sur 100 composantes principales pour tous les jeux de données et pour toutes les méthodes de représentation. La variance expliquée par ces axes est comprise entre 69,8% et 89,3%, en fonction des représentations utilisées.

5 Résultats

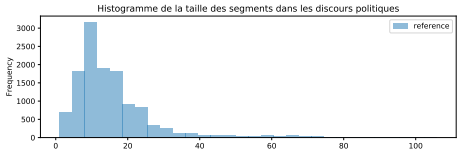
Etape 1 : Comparaison de méthodes neuronales et de méthodes statistiques usuelles. La Table 2 présente les résultats obtenus pour la segmentation de texte sur le corpus (discours politiques et textes juridiques). Les plongements lexicaux permettent une amélioration nette des performances par rapport à des méthodes statistiques usuelles. En comparant la segmentation de référence dans les corpus de discours politiques (cf. Figures 2a et 2b) et de textes juridiques (cf. Figures 3a et 3b), on s'aperçoit que les segmentations sont très différentes. En effet, les coupures permettent d'obtenir des blocs plus homogènes dans les discours politiques que dans les textes juridiques. Etant donné que la méthode séquentielle divise les documents en blocs assez homogènes (cf. Figures 2c, 2d, 3c et 3d), cette méthode est la plus performante pour les discours politiques. La réduction de dimension a permis une légère amélioration des résultats. Ce résultat est intéressant car cela signifie que l'on peut gagner en temps de calcul et en puissance de calcul sur cette méthode en utilisant de plus petits vecteurs, sans dégrader les résultats obtenus. Contrairement à la méthode séquentielle, la méthode sémantique génère des segments moins homogènes, surtout avec l'optimisation gourmande (cf. Figures 2e, 2f, 3e et 3f). Sur les textes de lois, cette méthode permet d'obtenir les meilleures performances de segmentation. Enfin, les indicateurs P_k et WindowDiff (WD) sont concordants, contrairement à la F-Mesure qui, décrite en section (2.4), ne rend pas compte des similarités de segments obtenus mais uniquement des frontières brutes.

	Discours			Lois		
	P_k	**WD**	**F-Mesure**	P_k	**WD**	**F-Mesure**
Text Tiling	0,431	0,431	0,101	0,404	0,404	0,052
C99	0,325	0,614	**0,229**	0,565	0,565	0,083
Sémantique + OG	0,274	0,282	0,187	**0,095**	**0,095**	0,116
Sémantique + OD	0,328	0,282	0,183	0,122	0,097	**0,128**
Séquentiel	0,272	0,277	0,101	0,192	0,194	0,074
Séquentiel + ACP	**0,269**	**0,270**	0,103	0,268	0,188	0,096

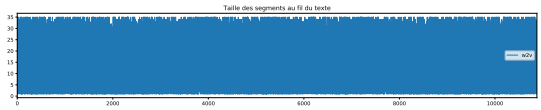
TABLE 2 – Résultats de segmentation obtenus sur les corpus de discours et les textes juridiques. Les méthodes ont été évaluées sur des fenêtres de taille 3-5 pour calculer P_k et WindowDiff et la plus petite erreur a été récupérée. OG : Optimisation Gourmande ; OD : Optimisation Dynamique



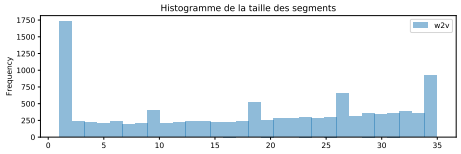
(a) Segments de référence



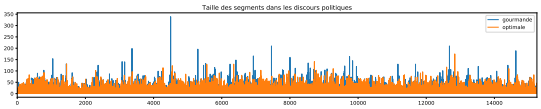
(b) Taille des segments de référence



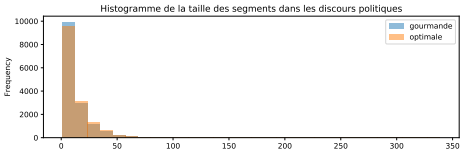
(c) Segments obtenus avec la méthode séquentielle



(d) Taille des segments avec la méthode séquentielle



(e) Segments obtenus avec la méthode sémantique



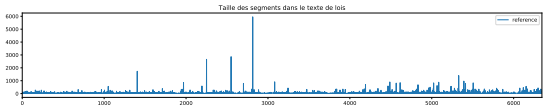
(f) Taille des segments avec la méthode sémantique

FIGURE 2 – Résultats obtenus sur les discours politiques

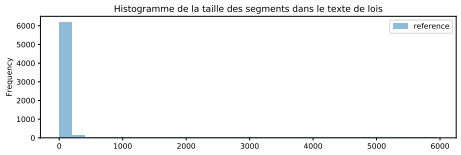
Etape 2 : Représentations contextuelles et non contextuelles. Ici, nous comparons l'utilisation de plongements lexicaux non contextuels (Word2Vec) et des plongements lexicaux contextuels (ELMo et USE) pour la segmentation de textes. L'utilisation de représentations vectorielles complexes n'a pas permis d'améliorer les résultats obtenus avec la méthode sémantique sur les textes juridiques (cf. Table 4). Quelle que soit la représentation utilisée, l'optimisation gourmande est plus performante que l'optimisation dynamique. *A contrario*, l'architecture transformer du modèle USE améliore les résultats obtenus avec la méthode séquentielle sur les discours politiques (cf. Table 3).

6 Conclusion

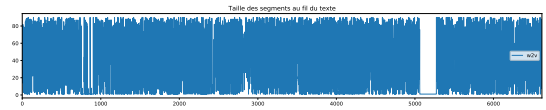
Dans cet article, nous nous sommes intéressés à la segmentation non supervisée de texte pour la détection de thématiques. Nous avons présenté un état de l'art des différentes méthodes utilisées pour



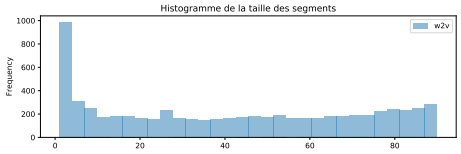
(a) Segments de référence



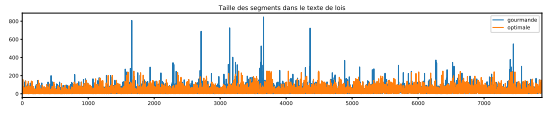
(b) Taille des segments de référence



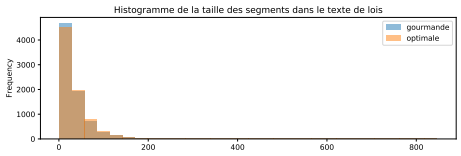
(c) Segments obtenus avec la méthode séquentielle



(d) Taille des segments avec la méthode séquentielle



(e) Segments obtenus avec la méthode sémantique



(f) Taille des segments avec la méthode sémantique

FIGURE 3 – Résultats obtenus sur les textes juridiques

	P_k	**WD**	**F-Mesure**
w2v	0,272	0,277	0,101
w2v + ACP	0,269	0,270	**0,103**
ELMo	0,273	0,276	0,098
ELMo + ACP	0,273	0,271	0,097
USE	**0,239**	**0,239**	0,067
USE + ACP	0,273	0,274	0,093

TABLE 3 – Résultats obtenus avec la méthode séquentielle sur les discours politiques.

	P_k	**WD**	**F-Mesure**
OG + w2v	**0,095**	**0,095**	0,116
OD + w2v	0,122	0,097	**0,128**
OG + ELMo	0,113	0,113	0,076
OD + ELMo	0,148	0,117	0,067
OG + USE	0,101	0,101	0,109
OD + USE	0,135	0,106	0,088

TABLE 4 – Résultats obtenus avec la méthode sémantique sur les textes juridiques.

cette tâche, en particulier pour leur application à des données spécifiques. Plus particulièrement, nous avons évalué l'impact des plongements lexicaux sur la performance des méthodes de segmentation. Nous avons vu que ces méthodes permettent une amélioration nette des résultats en comparaison avec des méthodes statistiques usuelles en segmentation automatique. Ensuite, nous nous sommes intéressés à la performance de deux algorithmes en fonction de la représentation des mots dans le corpus, contextuelle et non contextuelle. La méthode de calcul de plongements lexicaux n'a pas eu d'impact important sur le calcul des frontières de segmentation et semble dépendre de la méthode elle-même. Dans la suite de ce premier travail, nous allons nous intéresser à la mise au point d'une méthode basée sur des plongements de mots en essayant de contourner l'utilisation d'une fenêtre de calcul. Cela est un défaut de la méthode car fixer une fenêtre nous force à calculer un certain nombre de segments figé. Une fois les plongements lexicaux calculés, notre objectif serait de formuler une fonction objectif qui permettrait de maximiser la similarité au sein des phrases d'un segment et entre les segments. Enfin, nous souhaiterions appliquer la méthode de segmentation à des tâches de QA, et d'évaluer l'impact de notre méthode en recherche d'information.

Références

ALEMI A. A. & GINSPARG P. (2015). Text segmentation based on semantic word embeddings. *arXiv preprint arXiv :1503.05543*.

ANGHELUTA R., DE BUSSER R. & MOENS M.-F. (2002). The use of topic segmentation for automatic summarization. In *Proceedings of the ACL-2002 Workshop on Automatic Summarization*, p. 11–12.

BAEZA-YATES R., RIBEIRO-NETO B. *et al.* (1999). *Modern information retrieval*, volume 463. ACM press New York.

BEEFERMAN D., BERGER A. & LAFFERTY J. (1997). Text segmentation using exponential models. *arXiv preprint cmp-lg/9706016*.

BEEFERMAN D., BERGER A. & LAFFERTY J. (1999). Statistical models for text segmentation. *Machine learning*, **34**(1-3), 177–210.

CER D., YANG Y., KONG S.-Y., HUA N., LIMTIACO N., JOHN R. S., CONSTANT N., GUAJARDO-CESPEDES M., YUAN S., TAR C. *et al.* (2018). Universal sentence encoder. *arXiv preprint arXiv :1803.11175*.

CHE W., LIU Y., WANG Y., ZHENG B. & LIU T. (2018). Towards better UD parsing : Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 55–64, Brussels, Belgium : Association for Computational Linguistics.

CHO K., VAN MERRIËNBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*.

CHOI F. Y. (2000). Advances in domain independent linear text segmentation. *arXiv preprint cs/0003083*.

CHUANG W. T. & YANG J. (2000). Extracting sentence segments for text summarization : a machine learning approach. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, p. 152–159.

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*.

FAUCONNIER J.-P. (2015). French word embeddings.

GLAVAŠ G., NANNI F. & PONZETTO S. P. (2016). : Association for Computational Linguistics.

HALLIDAY M. & HASAN R. (1976). 1976 : Cohesion in english. london : Longman.

HEARST M. A. (1997). Texttiling : Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, **23**(1), 33–64.

HEITZ T., AZÉ J., ROCHE M., MELA A., PEINL P. & MEZAOUR A.-D. (2006). Présentation de deft 06 (dÉfi fouille de textes).

HUANG X., PENG F., SCHUURMANS D., CERCONE N. & ROBERTSON S. E. (2003). Applying machine learning to text segmentation for information retrieval. *Information Retrieval*, **6**(3-4), 333–362.

HURAULT-PLANTET M., JARDINO M. & BERTHELIN J.-B. (2006). Ajustement des frontières de segments thématiques détectés automatiquement. *Actes du 2ème dé fouilles de textes (DEFT), Fribourg, Suisse*.

KARUS K. (2019). Using embeddings to improve text segmentation.

KOSHOREK O., COHEN A., MOR N., ROTMAN M. & BERANT J. (2018). Text segmentation as a supervised learning task. *arXiv preprint arXiv :1803.09337*.

MAKHOUL J., KUBALA F., SCHWARTZ R., WEISCHEDEL R. *et al.* (1999). Performance measures for information extraction. In *Proceedings of DARPA broadcast news workshop*, p. 249–252 : Herndon, VA.

MARCU D. (2000). *The theory and practice of discourse parsing and summarization*. MIT press.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

MU J., STEGMANN K., MAYFIELD E., ROSÉ C. & FISCHER F. (2012). The acodea framework : Developing segmentation and classification schemes for fully automatic analysis of online discussions. *International journal of computer-supported collaborative learning*, **7**(2), 285–305.

PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

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*.

PEVZNER L. & HEARST M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, **28**(1), 19–36.

POLANYI L., CULY C., VAN DEN BERG M., THIONE G. L. & AHN D. (2004). Sentential structure and discourse parsing. In *Proceedings of the Workshop on Discourse Annotation*, p. 80–87.

POURVALI M. & ABADEH P. D. M. S. (2012). A new graph based text segmentation using wikipedia for automatic text summarization. *International Journal of Advanced Computer Science and Applications (IJACSA)*, **3**(1).

PRINCE V. & LABADIÉ A. (2007). Text segmentation based on document understanding for information retrieval. In *International Conference on Application of Natural Language to Information Systems*, p. 295–304 : Springer.

PURVER M., GRIFFITHS T. L., KÖRDING K. P. & TENENBAUM J. B. (2006). Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, p. 17–24 : Association for Computational Linguistics.

REYNAR J. C. (1998). Topic segmentation : Algorithms and applications.

SALTON G., SINGHAL A., BUCKLEY C. & MITRA M. (1996). Automatic text decomposition using text segments and text themes. In *Proceedings of the the seventh ACM conference on Hypertext*, p. 53–65.

SONG Y., MOU L., YAN R., YI L., ZHU Z., HU X. & ZHANG M. (2016). Dialogue session segmentation by embedding-enhanced texttiling. *arXiv preprint arXiv :1610.03955*.

UTIYAMA M. & ISAHARA H. (2001). A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, p. 499–506.

VAN RIJSBERGEN C. J. (1979). Information retrieval.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Spécificités des erreurs d'orthographe des personnes dyslexiques : analyse d'un corpus de productions écrites

Johana Bodard
Laboratoire CHArt, 2 rue de la Liberté, Saint-Denis, France
johana.bodard@etud.univ-paris8.fr

RÉSUMÉ

Cet article présente un travail d'analyse des erreurs d'orthographe de personnes dyslexiques à partir de corpus écrits en langue française. L'objectif de cette analyse est d'étudier la fréquence et les caractéristiques des erreurs afin de guider le développement de modules de correction orthographique spécifiques. Les résultats de cette analyse sont comparés aux travaux déjà réalisés en français, anglais et espagnol.

ABSTRACT

What are the characteristics of spelling errors made by dyslexics: an analysis of errors based on written corpora

In this paper, we present an analysis of spelling errors made by French dyslexics based on written corpora. The objective of this analysis is to investigate the frequency and characteristics of the spelling errors in order to guide the development of specific spell checking modules. The results of this analysis are compared with similar works in French, English and Spanish.

MOTS-CLÉS : dyslexie, analyse de corpus, correction orthographique.

KEYWORDS: dyslexia, corpus analysis, spell checking.

1 Introduction

La dyslexie est un trouble spécifique des apprentissages affectant le langage écrit dont la prévalence en France est estimée entre 6 et 8 % (Barrouillet *et al.*, 2007). Ce trouble entraine notamment des difficultés importantes dans l'acquisition de l'orthographe (dysorthographie), difficultés qui persistent souvent à l'âge adulte (Mazur-Palandre, 2018). Le correcteur orthographique apparait comme un outil particulièrement adapté pour pallier les difficultés orthographiques des personnes dyslexiques. Cependant, les correcteurs orthographiques classiques s'avèrent peu performants sur les écrits des dyslexiques (Bacquelé, 2015; Antoine *et al.*, 2019). Parmi les hypothèses avancées pour expliquer les faibles performances des correcteurs classiques sur ce type d'écrits, on peut citer : l'impossibilité de certains dyslexiques d'écrire correctement les initiales des mots (Bacquelé, 2015), le nombre important d'erreurs produisant des mots présents dans le dictionnaire (Antoine *et al.*, 2019), le nombre élevé d'erreurs par mot (Antoine *et al.*, 2019), la présence de mots mal découpés (fusionnés ou fragmentés) (Antoine *et al.*, 2019; Sitbon *et al.*, 2007), une écriture fortement phonétique (Sitbon *et al.*, 2007).

Afin de mettre en place des algorithmes de correction orthographique adaptés aux écrits des personnes

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 15–28.
HAL : hal-02786183.

dyslexiques, nous avons réalisé un travail préalable d'analyse des erreurs d'orthographe à partir de corpus écrits. Nous avons extrait et annoté les erreurs d'orthographe afin de vérifier les hypothèses citées dans le paragraphe précédent et de comparer les résultats obtenus à ceux des quelques travaux existant sur le sujet. Notre objectif est de guider les choix algorithmiques que nous opérerons lors du développement des modules de correction orthographique.

Il y a peu d'études s'intéressant aux troubles orthographiques des dyslexiques par rapport aux études s'intéressant à leurs difficultés en lecture, même si l'on constate un accroissement du nombre de recherches sur l'orthographe des dyslexiques depuis quelques années (Cidrim & Madeiro, 2017). Pourtant les difficultés en orthographe des personnes dyslexiques persistent davantage que leurs difficultés en lecture. Dans une étude comparant les performances en orthographe d'étudiants francophones dyslexiques et non dyslexiques de même âge et de même niveau scolaire, (Mazur-Palandre, 2018) constate des profils d'erreurs similaires entre les deux groupes d'étudiants : les dyslexiques font les mêmes types d'erreurs que les non dyslexiques dans les mêmes proportions relatives. Cependant, les étudiants dyslexiques font significativement plus d'erreurs que les étudiants non dyslexiques. De plus, une analyse qualitative des erreurs révèle que les étudiants dyslexiques font des erreurs atypiques qui ne sont jamais retrouvées dans les écrits des étudiants non dyslexiques, notamment en ce qui concerne les accords et la conjugaison (ex : *les personnes proviennes*, *j'ai préférez*).

Les travaux sur la constitution et l'analyse de corpus écrits ont surtout pour objectif d'étudier l'apprentissage d'une langue étrangère (Granger, 2009) ou l'apprentissage de l'écrit dans la langue maternelle (Wolfarth *et al.*, 2016). L'exploration de corpus de productions écrites de dyslexiques pour le développement de correcteurs orthographiques concernent peu de travaux : (Pedler, 2007) pour l'anglais, (Rello *et al.*, 2012, 2014) pour l'espagnol, et (Antoine *et al.*, 2019) pour le français.

Dans un premier temps, nous présenterons un état de l'art de l'analyse des erreurs à partir de corpus écrits pour la correction orthographique. Puis nous décrirons les corpus de textes à notre disposition et la méthodologie utilisée pour leur analyse. Nous décrirons ensuite les résultats de l'analyse et les comparerons aux travaux déjà réalisés. Enfin, nous conclurons sur les implications des résultats sur le développement de modules de correction orthographique.

2 État de l'art

(Damerau, 1964) propose quatre type d'erreurs simples pour la correction orthographique de mots isolés (c'est-à-dire, sans prise en compte du contexte) : l'insertion d'un caractère, l'omission d'un caractère, la substitution d'un caractère par un autre, la transposition de deux caractères adjacents. Il trouve que plus de 80 % des mots qui ne sont pas dans un dictionnaire diffèrent du mot attendu d'une seule erreur de l'un de ces quatre types. Ces travaux ont abouti au développement de la distance d'édition de Damerau-Levenshtein permettant de calculer le nombre minimum d'opérations nécessaires pour transformer une chaine de caractères en une autre chaine et sont utilisés pour la correction orthographique de mots isolés.

Ce taux de 80 % d'erreurs à une distance d'édition de 1 de leur forme correcte n'est pas retrouvé par (Mitton, 1987) sur un corpus de productions d'élèves faibles en orthographe. Seulement 69 % des erreurs issues de ce corpus entrent dans les quatre catégories d'erreurs définies par Damerau. Les autres erreurs sont en majorité des errreurs non lexicales (*real-word error*), c'est-à-dire des erreurs qui aboutissent à un mot qui existe dans le dictionnaire et qui ne peuvent donc pas être détectées

et corrigées sans prise en compte du contexte environnant. Concernant les écrits des personnes dyslexiques, on peut s'attendre à ce que le taux d'erreurs à une distance d'édition supérieure à 1 de leur forme correcte et le taux d'erreurs non lexicales soient encore plus importants.

Quelques travaux se sont intéressés à l'analyse des erreurs produites par les personnes dyslexiques pour la correction orthographique.

(Pedler, 2007) a constitué un corpus de productions écrites de personnes dyslexiques en langue anglaise. Un premier échantillon de 3134 mots dont 636 sont erronés (20 % du corpus) a été analysé avec la typologie suivante :

- erreur simple : une seule opération d'édition est nécessaire (parmi les 4 opérations définies par Damerau) pour passer du mot erroné au mot attendu
- erreur multiple : plus d'une opération d'édition sont nécessaires pour passer du mot erroné au mot attendu
- erreur de segmentation : fusion (omission d'un espace) ou segmentation (insertion d'un espace)

Cet échantillon contient 53 % d'erreurs simples, 39 % d'erreurs multiples et 8 % d'erreurs de segmentation. Les erreurs non lexicales représentent 17 % des erreurs du corpus. Cette première analyse montre que les personnes dyslexiques anglophones font beaucoup d'erreurs multiples et que le nombre d'erreurs non lexicales qu'ils produisent n'est pas négligeable.

Par la suite, (Pedler, 2007) s'est intéressée en particulier aux erreurs non lexicales. Elle a constitué un second corpus rassemblant des documents d'origines diverses (devoirs à la maison d'élèves, rédactions d'étudiants, expérimentation de saisie de texte en ligne, forums de discussion et listes de diffusion sur Internet, etc.) pour développer et évaluer un correcteur orthographique dédié à la correction de ce type d'erreurs. Ce corpus contient 21 524 mots dont 2654 sont erronés. Les erreurs non lexicales représentent près d'un tiers des erreurs de ce second corpus. Pour détecter et corriger ce type d'erreurs, elle propose de construire une liste de plusieurs milliers d'ensembles de confusion (ensembles de mots souvent confondus comme *loose* et *lose*) combinée à une analyse syntaxique et sémantique pour déterminer quel mot dans l'ensemble de confusion est le plus probable dans le contexte. Cependant, elle exclut les erreurs d'accord et de conjugaison des ensembles de confusion et ne peut donc pas corriger toutes les erreurs non lexicales avec cette approche (plus d'un tiers des erreurs non détectées sont des erreurs d'accord et de conjugaison).

(Rello *et al.*, 2012) ont constitué le premier corpus de productions écrites de personnes dyslexiques en langue espagnole (castillan), DysCorpus. Ce corpus comprend 16 textes manuscrits écrits par des enfants dyslexiques de 6 à 15 ans. Il contient 1057 mots dont 157 sont erronés (15 % du corpus). En reprenant la méthodologie utilisée par (Pedler, 2007), ils trouvent 67 % d'erreurs simples, 23 % d'erreurs multiples et 10 % d'erreurs de segmentation. Les erreurs non lexicales représentent 21 % des erreurs du corpus. Les auteurs expliquent le plus faible taux d'erreurs multiples en espagnol par rapport à l'anglais par le fait que l'orthographe de l'espagnol est plus transparente que celle de l'anglais. Cependant, le taux d'erreurs non lexicales est similaire dans les deux langues. Cela confirme que ce type d'erreurs constitue un véritable problème pour la correction orthographique des écrits des personnes dyslexiques.

(Rello *et al.*, 2014) ont étendu ce corpus avec de nouveaux textes pour atteindre un corpus de 83 textes manuscrits également rédigés par des enfants dyslexiques de 6 à 15 ans. Ils ont extrait de ce nouveau corpus 887 mots erronés et 1171 erreurs dans une liste, DysList, qu'ils ont enrichi de nombreuses informations linguistiques : distance d'édition, fréquence, longueur, position de l'erreur, nombre de syllabes et structure syllabique, type d'erreur reprenant la typologie de Damerau, erreur lexicale

ou non lexicale, informations visuelles (ex : lettres miroirs), informations phonétiques (comme le voisement ou le point d'articulation des phonèmes), transfert linguistique chez les enfants bilingues (catalan/castillan). Les travaux réalisés sur ce corpus sont utilisés pour la création d'un correcteur orthographique (Rello *et al.*, 2015) pour les dys en langue espagnole. Appliqué à la détection et à la correction des erreurs non lexicales, ce correcteur détecte et corrige plus d'erreurs non lexicales que les correcteurs classiques, mais au prix d'une précision moindre (plus de faux positifs).

En langue française, (Antoine *et al.*, 2019) ont constitué un corpus de textes rédigés par 5 enfants dyslexiques et 5 enfants paralysés cérébraux pour un système d'aide à la communication combinant prédiction et correction orthographique. Ce corpus rassemble 521 erreurs orthographiques qui ont été annotées en suivant un schéma d'annotation répondant aux besoins des chercheurs en TAL et à ceux des orthophonistes. Pour la recherche en TAL, ils notent si le mot comporte une ou plusieurs erreurs distinctes, le type d'erreur (lexicale, syntaxique ou sémantique) et la morphologie en distinguant les erreurs de segmentation (fragmentation ou fusion) des autres erreurs pour lesquelles ils calculent la distance d'édition de Damerau-Levenshtein entre la forme erronée et la forme attendue. Pour les besoins des orthophonistes, ils établissent une typologie des erreurs en distinguant les erreurs phonologiquement plausibles (erreurs qui ne modifient pas la prononciation du mot, par exemple : *insi* (*ainsi*)) et les erreurs phonologiquement non plausibles (erreurs qui modifient la prononciation du mot, par exemple : *cantre* (*centre*)).

Leurs travaux, les premiers s'intéressant à la langue française, montrent un taux d'erreurs multiples de 54 % similaire à celui retrouvé par (Pedler, 2007) en anglais. Cependant, les taux d'erreurs non lexicales (29 %) et d'erreurs de segmentation (15 %) sont supérieurs à ceux retrouvés en anglais et en espagnol. Cela suggère que les problèmes rencontrés chez les dyslexiques anglais et espagnols sont également retrouvés, dans des proportions plus importantes, chez les dyslexiques français.

3 Méthodologie d'analyse des corpus

3.1 Description des corpus

Pour cette étude, nous avons utilisé deux corpus de productions écrites de personnes dyslexiques. Le premier corpus nous a été fourni par la FFDys [1], le second par une orthophoniste qui travaille avec des personnes dyslexiques en lien avec la FFDys.

Le premier corpus contient 9 textes scolaires (contrôles, exercices, dictées) écrits par des élèves dyslexiques de collèges et lycées (de la 5e à la terminale). Sept textes ont été écrits au clavier, les deux autres sont des textes manuscrits. Ils ont été écartés de la présente étude. En effet, le mode d'entrée du texte peut avoir un impact sur le type d'erreurs produites. (Sitbon *et al.*, 2007) constatent que certaines erreurs rencontrées dans les textes manuscrits d'enfants dyslexiques, telles que les substitutions de lettres miroirs (p/q ou b/d par exemple), ne sont pas observées sur des textes écrits au clavier. De plus, l'utilisation du clavier entraine des erreurs de frappe qu'on ne retrouvera pas dans les textes manuscrits. Les 7 textes écrits au clavier totalisent 3357 mots [2]. Ce sont des textes relativement longs (475 mots en moyenne par texte). Ce premier corpus contient 1240 formes erronées [3] dont 771 formes

1. Fédération Française des Dys
2. Nous entendons par mot toute séquence de caractères séparée par des espaces ou de la ponctuation.
3. Nous préférons parler de formes erronées plutôt que de mots erronés. Une forme erronée peut correspondre à un ou plusieurs mots. Ex : *plus par* (*plupart*)

distinctes.

Le second corpus contient 71 textes courts (53 mots en moyenne par texte) écrits au clavier par des personnes dyslexiques âgées de 16 à 45 ans (âge moyen = 22,5 ans, écart-type = 4,7 ans). Ce corpus est lui-même composé de :

— 6 dictées
— 33 expressions écrites dirigées
— 32 expressions écrites libres

Il totalise 3913 mots et 879 formes erronées dont 594 formes distinctes.

L'ensemble des deux corpus totalisent 7270 mots et 2119 formes erronées dont 1303 distinctes.

3.2 Annotation des erreurs

Pour chaque texte, nous avons extrait manuellement les formes erronées dans un tableau, puis pour chaque forme erronée, nous avons noté :

— la forme erronée
— la forme attendue
— le lemme [4] de la forme attendue
— la phrase contenant la forme erronée
— le nombre d'erreurs et leurs types
— la distance d'édition de Damerau-Levenshtein entre la forme erronée et la forme attendue
— la similarité entre les transcriptions phonétiques des formes erronée et attendue
— si l'erreur est lexicale ou non-lexicale
— le nombre de mots erronés dans le contexte (les 2 mots précédents et les 2 mots suivants)

3.3 Les différents types d'erreurs

Au lieu de distinguer comme (Pedler, 2007) et (Rello *et al.*, 2012) les erreurs simples et les erreurs multiples, nous calculons, d'une part, la distance d'édition entre la forme erronée et la forme attendue et, d'autre part, nous comptons le nombre d'erreurs de la forme erronée comme (Antoine *et al.*, 2019).

Pour les types d'erreurs, nous utilisons la typologie présentée dans la table 1. Cette typologie s'inspire de celle définie par (Plisson & Daigle, 2013) pour décrire les erreurs d'enfants dyslexiques francophones. Par rapport à cette typologie, nous ne distinguons pas les erreurs phonologiquement plausibles des erreurs non phonologiquement plausibles. Nous regroupons dans une même catégorie les erreurs de phonétisation concernant les mauvais choix de graphèmes [5], les lettres muettes et les morphogrammes lexicaux [6]. L'idée étant que ces différents types d'erreurs peuvent être corrigés avec la même approche. De même, nous comptabilisons les erreurs concernant les traits d'union dans les erreurs de segmentation plutôt qu'avec les erreurs sur les majuscules.

4. Le lemme d'un mot est sa forme canonique telle qu'on la trouve dans un lexique. Préciser le lemme permet de distinguer les homographes tels que *est* forme conjuguée du lemme *être* et *est* point cardinal).

5. (Catach, 1986) définit le graphème comme la plus petite unité distinctive de la chaine écrite et le phonème comme la plus petite unité distinctive de la chaine orale. Par exemple, le mot *châteaux* se décompose en 5 graphèmes : ‹ ch ›, ‹ â ›, ‹ t ›, ‹ eau › et ‹ x › et en 4 phonèmes : /ʃ/, /ɑ/, /t/ et /o/.

6. (Catach, 1986) définit le morphogramme lexical comme un graphème non chargé de transcrire un phonème et permettant d'établir un lien avec les dérivés. Par exemple, le ‹ t › final dans *petit*.

En fonction de son type, une erreur peut concerner l'ensemble d'un mot (ex : confusion entre les homophones *ces* et *c'est*), un graphème (ex : substitution du graphème ‹ss› par le graphème ‹s› dans *réusite*) ou un caractère (ex : omission de l'apostrophe dans *lafrique*).

Type d'erreurs	**Exemples**
Phonétisation : mauvais choix de graphème et lettre muette	comerse (commerce), toujour (toujours)
Substitution d'un graphème par un autre phonétiquement proche	réusite (réussite)
Confusion entre homophones	ces (c'est)
Erreur d'accord en genre et nombre et de conjugaison	autre (autres), rajouterai (rajouterait)
Erreur de segmentation : fragmentation ou fusion (incluant les erreurs concernant les apostrophes et les traits d'union)	quel que (quelque), ducou (du coup), lafrique (l'Afrique), rendévous (rendez-vous)
Liaison erronée	on na (on a)
Majuscule	japon (Japon)
Ajout d'un caractère	situiation (situation)
Omission d'un caractère	Qustion (Question)
Substitution d'un caractère par un autre caractère	dont (sont)
Transposition de deux caractères adjacents	aprle (parle)
Déplacement d'un caractère	disgetif (digestif)
Omission ou répétition de mot	il trouve pas (il ne trouve pas)
Mauvais choix lexical	famille (familiale)
Mot non reconnu	sanéte

TABLE 1: Types d'erreurs

4 Résultats

4.1 Distance de Damerau-Levenshtein

La table 2 présente les pourcentages de formes erronées à une distance de 1, 2 ou plus de leur forme correcte pour chaque corpus et pour les deux corpus. En moyenne sur les deux corpus, une large proportion de formes erronées (41 %) sont à une distance de 2 ou plus de leur forme correcte. On note cependant une différence importante entre les deux corpus : sur le premier corpus, un peu moins de la moitié des formes sont concernées, un tiers des formes sur le second corpus.

Dans le premier corpus, la distance maximum est de 7 et concerne deux formes erronées : *oré* (*auraient*) et *nalé* (*n'allaient*). Dans le deuxième corpus, la distance maximum est de 5 et concerne trois formes erronées : *fesé* (*faisais*), *noyer* (*nettoyé*) et *setoufle* (*s'étouffent*).

Corpus	Distance = 1	Distance = 2	Distance > 2
1	53 %	24,7 %	22,3 %
2	67,1 %	21,8 %	11,1 %
1 et 2	58,8 %	23,5 %	17,7 %

TABLE 2: Distance de Damerau-Levenshtein

4.2 Similarité phonétique

Dans un premier temps, nous avons comparé la transcription phonétique des formes erronées et attendues. Ces transcriptions ont été obtenues grâce au transcripteur LIA_PHON (Béchet, 2001). Puis, nous avons comparé les phonétiques après simplification de la phonétique des voyelles. Nous avons réduit le nombre de voyelles prises en compte par LIA_PHON de 15 à 10 : nous ne distinguons plus les voyelles /e/ et /ɛ/ (dans *thé* et *cette*), /o/ et /ɔ/ (dans *tôt* et *botte*), /ø/, /œ/ et /ə/ (dans *peu*, *peur* et *le*), et /ɛ̃/ et /œ̃/ (dans *brin* et *brun*). En effet, suivant la personne ou la région, la prononciation des voyelles peut varier (par exemple, *très* est prononcé [tʁe] ou [tʁɛ]) et certaines oppositions peuvent disparaitre (pas de distinction entre *brin* et *brun* par exemple).

La table 3 présente les pourcentages de formes erronées ayant la même phonétique que leur forme correcte (c'est-à-dire, les erreurs phonologiquement plausibles) pour chaque corpus et pour les deux corpus. Dans les deux corpus, plus de la moitié des formes erronées ont une phonétique identique à celle de leur forme correcte. Si on utilise la phonétique simplifiée, deux tiers des formes erronées ont une phonétique proche de celle de leur forme correcte.

Corpus	Phonétique	Phonétique simplifiée
1	58,9 %	69,7 %
2	58,5 %	63,4 %
1 et 2	58,7 %	67,1 %

TABLE 3: Similarité phonétique

4.3 Erreurs non lexicales

Une erreur non lexicale est une erreur qui produit un mot présent dans le lexique. Il s'agit essentiellement d'erreurs syntaxiques (ex : *les région* au lieu de *les régions*) et sémantiques (ex : *famille* au lieu de *familial*). Plus rarement, les erreurs de segmentation peuvent produire des erreurs non lexicales (ex : *plus par* au lieu de *plupart*, *lest* au lieu de *l'est*).

Le choix du lexique qui sert à la correction orthographique est important. Plus celui-ci est large plus il va contenir des formes rares, peu usitées et plus le risque qu'une forme erronée se retrouve dans le lexique augmente. Par exemple : Les formes erronées *oré* (*auraient*), *mayeur* (*meilleur*) et *este* (*Est*) sont dans le lexique Morphalou 3. Pour détecter l'erreur, il faut alors utiliser une analyse syntaxique voire sémantique.

Nous avons comparé 3 lexiques :

— Morphalou (version 3.1) (ATILF, 2019) : un lexique à large couverture qui aggrège plusieurs lexiques pour atteindre 954 690 formes fléchies

— Dicollecte (version 6.4.1)[7] : un lexique de plus de 500 000 formes fléchies utilisé par le correcteur orthographique Hunspell en français
— Lexique (version 3.83) (New *et al.*, 2004) : un lexique de plus de 140 000 formes fléchies

La table 4 présentent les pourcentages d'erreurs non lexicales relevées dans les corpus en fonction du lexique choisi. Quel que soit le lexique utilisé, sur l'ensemble des deux corpus, un peu plus de la moitié des formes erronées sont des erreurs non lexicales. Même si le deuxième corpus contient une proportion plus faibles d'erreurs que le premier corpus, le pourcentage d'erreurs non lexicales y est plus élevé.

Corpus	Morphalou 3	Dicollecte	Lexique 3
1	607 (49 %)	595 (48 %)	587 (48 %)
2	531 (60 %)	523 (59 %)	523 (59 %)
1 et 2	1138 (54 %)	1118 (53 %)	1110 (53 %)

TABLE 4: Erreurs non lexicales

4.4 Formes correctes les plus souvent erronées

Les 10 formes correctes les plus fréquemment erronées sont des mots courts (1 à 5 lettres), le plus souvent monosyllabiques (à l'exception de *après* qui est constitué de 2 syllabes, ils possèdent tous 1 seule syllabe) et fréquents. La table 5 présente les 10 formes correctes les plus fréquemment erronées, le nombre d'occurrences erronées, le pourcentage de formes erronées et les différentes formes erronées.

Forme correcte	Nombre d'occurrences erronées	Pourcentage d'occurences erronées	Formes erronées
très	21	87,5 %	tré, tres
peut	13	86,7 %	pue, peu, pela
à	115	81,6 %	a, d, ∅
après	12	80 %	apres, apré, apra, apre, a prais, apret
ils	12	73,3 %	il
ont	13	72,2 %	on
c'est	21	58,3 %	ses, sé, ces, s'est, cces
ce	18	42,9 %	se, si
au	15	29,4 %	o, a
est	23	28,4 %	et, é, n'ait, ai, soi, ∅

TABLE 5: Formes les plus fréquemment erronées

4.5 Erreurs sur la première lettre

D'après (Yannakoudakis & Fawthrop, 1983), la première lettre d'un mot erroné est correcte dans la majorité des cas en anglais (moins de 2 % des erreurs sont retrouvées à l'initiale des mots).

7. Ce lexique est téléchargeable à l'adresse https ://grammalecte.net/download.php ?prj=fr

Dans l'ensemble de nos corpus, nous avons 16,5 % de formes dont la première lettre est erronée (10,9 % si on exclut les mots d'une seule lettre). On ne prend pas en compte les erreurs de majuscule.

Si l'on regarde la phonétique, moins de 4 % des formes erronées sont phonétiquement incorrectes à l'initiale.

4.6 Variabilité des formes erronées

Dans l'ensemble du corpus, 200 formes correctes ont au moins deux formes erronées (18,3 % des 1092 formes correctes distinctes).

On compte jusqu'à 6 formes erronées différentes dans l'ensemble du corpus pour la forme *après*.

4.7 Contexte autour du mot

La correction orthographique nécessite souvent une analyse contextuelle :

- pour les erreurs lexicales : pour sélectionner la meilleure correction parmi une liste de corrections potentielles
- pour les erreurs non lexicales : pour les détecter et les corriger

Cependant, si le contexte autour du mot est erroné, l'analyse contextuelle peut donner des résultats erronés.

Pour chaque mot erroné nous avons regardé, si le contexte local (2 mots avant et 2 mots après) était correct ou erroné. La table 6 présente les proportions de formes erronées avec aucun, un ou plusieurs mots erronés dans leur contexte. 72,3 % des formes erronées ont au moins un mot de contexte erroné.

Nombre de mots erronés	Pourcentage de formes erronées concernées
0	27,7 %
1	39,5 %
2	24,4 %
3	7,4 %
4	1,0 %

TABLE 6: Nombre de mots de contexte erroné

4.8 Répartitions des erreurs dans les différentes catégories

Le nombre moyen d'erreurs par mot est de 1,4 sur l'ensemble des corpus. Le premier corpus a en moyenne 1,48 erreurs par mot (écart-type = 0,13), le second corpus a en moyenne 1,24 erreurs par mot (écart-type = 0,19). Le nombre maximum d'erreurs par mot est de 5 pour le premier corpus et de 4 pour le second corpus. Une forme erronée peut combiner plusieurs erreurs de types différents. Par exemple, la forme erronée *meiu* (*mieux*) combine une transposition de caractères adjacents et une omission de lettre muette.

La répartition des erreurs en fonction de leur type est présentée dans la table 7. Les erreurs les plus fréquentes sont les erreurs de phonétisation et les erreurs d'accord en genre et nombre et de conjugaison. Ces deux catégories d'erreurs représentent plus de la moitié des erreurs.

Type d'erreurs	Exemples	Pourcentage d'erreurs
Phonétisation : mauvais choix de graphème et lettre muette	comerse (commerce), toujour (toujours)	27,25 %
Erreur d'accord en genre et nombre et de conjugaison	autre (autres), rajouterai (rajouterait)	26,81 %
Substitution d'un graphème par un autre phonétiquement proche	réusite (réussite)	15,98 %
Confusion entre homophones	ces (c'est)	11,28 %
Erreur de segmentation : fragmentation ou fusion (incluant les erreurs concernant les apostrophes et les traits d'union)	quel que (quelque), ducou (du coup), lafrique (l'Afrique), rendévous (rendez-vous)	6,35 %
Majuscule	japon (Japon)	3,14 %
Omission d'un caractère	Qustion (Question)	3,04 %
Substitution d'un caractère par un autre caractère	dont (sont)	1,59 %
Omission ou ajout de mot	il trouve pas (il ne trouve pas)	1,28 %
Ajout d'un caractère	situiation (situation)	1,25 %
Transposition de deux caractères adjacents	aprle (parle)	1,01 %
Mauvais choix lexical	famille (familiale)	0,54 %
Mot non reconnu	sanéte	0,17 %
Liaison erronée	on na (on a)	0,20 %
Déplacement d'un caractère	disgetif (digestif)	0,07 %

TABLE 7: Répartition des erreurs

5 Discussion des résultats

Les corpus que nous avons analysés contiennent environ un tiers de mots erronés. Ce taux est plus important que ceux trouvés dans les travaux de (Pedler, 2007) en anglais (20 % de mots erronés) et de (Rello *et al.*, 2012) en espagnol (15 % de mots erronés). Il est cependant plus faible que celui trouvé par (Antoine *et al.*, 2019) en français (55 % de mots erronés). Nous observons également un nombre moyen d'erreurs par mot inférieur à celui observé par (Antoine *et al.*, 2019) : 1,4 contre 1,8 erreurs par mot.

Concernant la distance d'édition, nous avons un taux d'erreurs multiples de 41,2 %, supérieur aux 23 % de (Rello *et al.*, 2012), mais proche des 39 % de (Pedler, 2007). De nouveau, (Antoine *et al.*, 2019) trouve un taux supérieur (54 %).

Nous avons également un taux d'erreurs de segmentation proche de celui observé par (Pedler, 2007) (respectivement 6 % et 8 %). (Rello *et al.*, 2012) et (Antoine *et al.*, 2019) trouvent des taux légèrement plus élevés (respectivement 10 % et 15 %). Même si le taux d'erreurs de segmentation que nous relevons est relativement faible, ces erreurs ne doivent pas être négligées car elles peuvent être particulièrement difficiles à corriger quand elles produisent des erreurs non lexicales (*laide* au lieu de

l'aide) ou qu'elles sont cumulées à d'autres erreurs (*apeures* au lieu de *à peu près*).

Les taux plus faibles que nous observons par rapport aux résultats de (Antoine *et al.*, 2019) peuvent s'expliquer par le fait que nous avons des textes rédigés par des adultes ou des élèves de collège et lycée alors que l'âge moyen des enfants dyslexiques de l'étude de (Antoine *et al.*, 2019) est de 10 ans. Nous observons également une différence entre notre premier corpus constitué d'écrits de collégiens et de lycéens et notre second corpus constitués d'écrits rédigés par des personnes plus âgées : les textes du premier corpus contiennent en moyenne plus de formes erronées, plus d'erreurs par mot et plus d'erreurs multiples. Cela pose la question de savoir quelles sont les compétences en orthographe qui peuvent être améliorées par l'apprentissage chez les personnes dyslexiques.

Nous observons un taux d'erreurs non lexicales nettement supérieur aux autres langues : plus de la moitié des formes erronées de nos corpus sont présentes dans les trois lexiques que nous avons testés, contre 17 % pour (Pedler, 2007) en anglais et 9 % pour (Rello *et al.*, 2014) en espagnol. Une analyse plus fine des erreurs non lexicales de nos corpus est nécessaire pour comprendre cette différence avec les autres langues, notamment pour connaitre la proportion d'erreurs syntaxiques et d'erreurs sémantiques parmi ces erreurs non lexicales. À titre d'exemple, (Antoine *et al.*, 2019) trouvent 29 % d'erreurs non lexicales dans leur corpus en français.

Étant donné le nombre très élevé d'erreurs non lexicales, une analyse contextuelle est indispensable. L'analyse de notre corpus montre que le contexte contient souvent des mots erronés ce qui peut perturber l'analyse contextuelle. Nous nous sommes pour l'instant contenter de regarder si le contexte local (les 2 mots précédents et les 2 mots suivants) était erroné. L'analyse du contexte peut être poussée plus loin en regardant la proportion d'erreurs pouvant être corrigées dans un contexte local et celles nécessitant une analyse plus globale comme la phrase, en examinant les performances des étiqueteurs morpho-syntaxiques sur des phrases issues de nos corpus ou encore en regardant si le contexte est phonétiquement correct.

Nous trouvons un taux d'erreurs sur la première lettre de 16 % similaire à celui retrouvé par (Antoine *et al.*, 2019) (14 %) et légèrement supérieur à ceux observés par (Rello *et al.*, 2012) et (Pedler, 2007) (respectivement 10 % et 5 %). Ce taux reste assez faible. Cela ne confirme donc pas l'hypothèse avancée par (Bacquelé, 2015) selon laquelle les dyslexiques auraient des difficultés pour écrire correctement les initiales des mots. De plus, il est possible de s'appuyer sur la phonétique qui est correcte sur la première lettre dans plus de 96 % des cas.

Le taux de formes erronées phonétiquement similaires à leur forme correcte est de 59 %. Ce résultat est conforme aux observations de (Antoine *et al.*, 2019) qui trouvent 62 % d'erreurs phonologiquement plausibles. Cela suggère que le passage à la phonétique peut être intéressant pour corriger certaines erreurs comme le proposent (Sitbon *et al.*, 2007). En simplifiant les règles de transcription en phonétique des voyelles, nous trouvons deux tiers d'erreurs phonétiquement similaires à leur forme erronée. Il faut cependant trouver un juste milieu : plus on simplifie les règles de transcription en phonétique, plus on augmente le nombre de mots du lexique phonétiquement similaires aux mots erronés.

Concernant la variabilité des formes erronées, notre analyse montre une variabilité inter-individuelle parfois importante. Cependant, nous n'avons pas pu analyser la variabilité intra-individuelle car nous ne disposons pas d'assez de textes écrits par une même personne. Cette question est toutefois intéressante : si une même personne dyslexique reproduit souvent le même type d'erreurs, il est peut-être possible de modéliser son profil d'erreurs.

Enfin, il serait également intéressant d'analyser plus précisément les difficultés communes à toutes

les personnes dyslexiques quelle que soit la langue et celles qui sont spécifiques à chaque langue. Par exemple, le français est une langue à l'orthographe plutôt opaque : les relations entres phonèmes et graphèmes sont irrégulières. En particulier, dans le sens de l'écriture, le français est proche de l'anglais, langue à l'orthographe très opaque. Une étude comparant les erreurs produites par des dyslexiques grecs et des dyslexiques américains suggère que les différentes caractéristiques d'une langue entrainent différents types et proportions d'erreurs : les dyslexiques grecs (dont la langue est considérée comme transparente) font en proportion significativement moins d'erreurs phonologiquement non plausibles que les dyslexiques américains, mais plus d'erreurs phonologiquement plausibles et plus d'erreurs grammaticales (Giannouli & Pavlidis, 2014). Une autre particularité du français est le décalage important entre la morphologie de l'oral et celle de l'écrit (Jaffré, 2005) : les marques de genre et de nombre, les terminaisons verbales sont très présentes à l'écrit, mais quasiment absentes à l'oral. Ce n'est pas le cas de l'anglais ou de l'espagnol et cela peut constituer une difficulté supplémentaire pour les dyslexiques francophones.

6 Conclusion et travaux futurs

Dans cet article, nous avons présenté une analyse des erreurs produites par des personnes dyslexiques françaises (collégiens, lycéens et adultes) à partir de corpus écrits. Comme (Antoine *et al.*, 2019), nous retrouvons des taux de mots erronés, d'erreurs multiples et d'erreurs non lexicales supérieurs à ceux observés dans des corpus de personnes dyslexiques anglaises ou espagnoles.

Cette étude soulève plusieurs points importants :

— le nombre très élevé d'erreurs non lexicales : une analyse contextuelle est indispensable pour détecter et corriger efficacement ces erreurs
— le nombre d'erreurs phonétiquement similaires ou proches de leur forme correcte étant important, dans quelle mesure peut-on s'appuyer sur la phonétique pour corriger les erreurs ?
— le contexte pouvant lui-même contenir des erreurs, dans quelle mesure peut-on s'appuyer sur le contexte pour détecter et corriger les erreurs ?

La prochaine étape consistera à évaluer les performances des correcteurs orthographiques actuels sur ce corpus pour identifier les erreurs qu'ils parviennent à corriger et celles qu'ils ne parviennent pas à détecter ou corriger. Nous disposons actuellement d'un premier module de correction basique fondé sur la phonétique et nous comparerons les résultats obtenus par notre module avec ceux des correcteurs.

Références

ANTOINE J.-Y., CROCHETET M., ARBIZU C., LOPEZ E., POUPLIN S., BESNIER A. & THEBAUD M. (2019). Ma copie adore le vélo : analyse des besoins réels en correction orthographique sur un corpus de dictées d'enfants. In *TALN 2019*, Toulouse, France. HAL : hal-02375246.

ATILF (2019). Morphalou. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

BACQUELÉ V. (2015). L'usage de l'informatique par les élèves dyslexiques : un outil de compensation à l'épreuve de l'inclusion scolaire. *Terminal. Technologie de l'information, culture & société*, (116). DOI : 10.4000/terminal.661.

BARROUILLET P., BILLARD C., AGOSTINI M. D., DÉMONET J.-F., FAYOL M., GOMBERT J.-E., HABIB M., NORMAND M.-T. L., RAMUS F., SPRENGER-CHAROLLES L. & VALDOIS S. (2007). *Dyslexie, dysorthographie, dyscalculie : bilan des données scientifiques*. Rapport de recherche, INSERM. HAL : hal-01570674.

BÉCHET F. (2001). LIA_phon : un système complet de phonétisation de textes. *Traitement Automatique des Langues*, **42**(1), 47–67.

CATACH N. (1986). *L'orthographe française : traité théorique et pratique : avec des travaux d'application et leurs corrigés (avec la collab. de Claude Gruaz et Daniel Duprez)*. Paris, nathan édition.

CIDRIM L. & MADEIRO F. (2017). Studies on spelling in the context of dyslexia : a literature review. *Revista CEFAC*, **19**(6), 842–854. DOI : 10.1590/1982-0216201719610317.

DAMERAU F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, **7**(3), 171–176. DOI : 10.1145/363958.363994.

GIANNOULI V. & PAVLIDIS G. T. (2014). What can spelling errors tell us about the causes and treatment of dyslexia ? : What can Spelling Errors Tell Us about the Causes and Treatment of Dyslexia ? *Support for Learning*, **29**(3), 244–260. DOI : 10.1111/1467-9604.12065.

GRANGER S. (2009). The contribution of learner corpora to second language acquisition and foreign language teaching. *Corpora and language teaching*, **33**, 13–32. DOI : 10.1075/scl.33.04gra.

JAFFRÉ J.-P. (2005). L'orthographe du français, une exception ? *Le francais aujourd'hui*, **n° 148**(1), 23–31. DOI : 10.3917/lfa.148.0023.

MAZUR-PALANDRE A. (2018). La dyslexie à l'âge adulte : la persistance des difficultés orthographiques. *SHS Web of Conferences*, **46**, 10003. DOI : 10.1051/shsconf/20184610003.

MITTON R. (1987). Spelling checkers, spelling correctors and the misspellings of poor spellers. *Information Processing & Management*, **23**(5), 495–505. DOI : 10.1016/0306-4573(87)90116-6.

NEW B., PALLIER C., BRYSBAERT M. & FERRAND L. (2004). Lexique 2 : A new French lexical database. *Behavior Research Methods, Instruments, & Computers*, **36**(3), 516–524. DOI : 10.3758/BF03195598.

PEDLER J. (2007). *Computer Correction of Real-word Spelling Errors in Dyslexic Text*. Thèse de doctorat, University of London.

PLISSON A. & DAIGLE, DANIEL ANDYA MONTESINOS-GELET I. (2013). The Spelling Skills of French-Speaking Dyslexic Children. *Dyslexia*, **19**(2), 76–91. DOI : 10.1002/dys.1454.

RELLO L., BAEZA-YATES R. & LLISTERRI J. (2014). DysList : An Annotated Resource of Dyslexic Errors. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, p. 1289–1296, Reykjavik, Iceland : European Language Resources Association (ELRA). DOI : 10.13140/2.1.2542.7205.

RELLO L., BAEZA-YATES R., SAGGION H. & PEDLER J. (2012). A First Approach to the Creation of a Spanish Corpus of Dyslexic Texts. In *LREC Workshop Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, p. 22–27, Turkey.

RELLO L., BALLESTEROS M. & BIGHAM J. P. (2015). A Spellchecker for Dyslexia. p. 39–47 : ACM. DOI : 10.1145/2700648.2809850.

SITBON L., BELLOT P. & BLACHE P. (2007). Traitements phrastiques phonétiques pour la réécriture de phrases dysorthographiées. In *TALN 2007*, Toulouse, France. HAL : hal-01321119.

WOLFARTH C., PONTON C. & BRISSAUD C. (2016). Du TAL dans les écrits scolaires : premières approches. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2016*, volume 09, p. 30–37, Paris. HAL : hal-01878718.

YANNAKOUDAKIS E. J. & FAWTHROP D. (1983). The rules of spelling errors. *Information Processing & Management*, **19**(2), 87–99. DOI : 10.1016/0306-4573(83)90045-6.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Ré-entraîner ou entraîner soi-même ? Stratégies de pré-entraînement de BERT en domaine médical

Hicham El Boukkouri
Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France
hicham.elboukkouri@limsi.fr

RÉSUMÉ

Les modèles BERT employés en domaine spécialisé semblent tous découler d'une stratégie assez simple : utiliser le modèle BERT originel comme initialisation puis poursuivre l'entraînement de celui-ci sur un corpus spécialisé. Il est clair que cette approche aboutit à des modèles plutôt performants (e.g. BioBERT (Lee *et al.*, 2020), SciBERT (Beltagy *et al.*, 2019), BlueBERT (Peng *et al.*, 2019)). Cependant, il paraît raisonnable de penser qu'entraîner un modèle directement sur un corpus spécialisé, en employant un vocabulaire spécialisé, puisse aboutir à des plongements mieux adaptés au domaine et donc faire progresser les performances. Afin de tester cette hypothèse, nous entraînons des modèles BERT à partir de zéro en testant différentes configurations mêlant corpus généraux et corpus médicaux et biomédicaux. Sur la base d'évaluations menées sur quatre tâches différentes, nous constatons que le corpus de départ influence peu la performance d'un modèle BERT lorsque celui-ci est ré-entraîné sur un corpus médical.

ABSTRACT

Re-train or train from scratch ? Pre-training strategies for BERT in the medical domain

BERT models used in specialized domains all seem to be the result of a simple strategy : initializing with the original BERT then resuming pre-training on a specialized corpus. This method yields rather good performance (e.g. BioBERT (Lee *et al.*, 2020), SciBERT (Beltagy *et al.*, 2019), BlueBERT (Peng *et al.*, 2019)). However, it seems reasonable to think that training directly on a specialized corpus, using a specialized vocabulary, could result in more tailored embeddings and thus help performance. To test this hypothesis, we train BERT models from scratch using many configurations involving general and medical corpora. Based on evaluations using four different tasks, we find that the initial corpus only has a weak influence on the performance of BERT models when these are further pre-trained on a medical corpus.

MOTS-CLÉS : plongements de mots, plongements contextualisés, BERT, domaine médical, biomédical, domaine spécialisé, adaptation au domaine.

KEYWORDS: word embeddings, contextualized embeddings, BERT, medical domain, biomedical, specialized domain, domain adaptation.

1 Introduction

Les dernières années ont été témoins de l'apparition de nombreuses approches d'apprentissage par transfert en traitement automatique des langues (TAL). Ayant initialement connu un grand succès en traitement de la parole (Wang & Zheng, 2015) et en vision par ordinateur (He *et al.*, 2017), ces

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 29–42.
HAL : hal-02786184.

approches ont rapidement trouvé leur application en TAL avec notamment ULMFiT (Howard & Ruder, 2018) qui a pu montrer l'efficacité des modèles de langue pré-entraînés une fois adaptés pour des tâches de classification de texte. Peu de temps après, Radford *et al.* (2018) ont étendu ces résultats à d'autres tâches classiques du TAL (implicature textuelle, compréhension de texte…).

Suite à l'engouement autour de l'apprentissage par transfert, les modèles de plongement de mots ont eux aussi connu une importante rupture, avec l'apparition de modèles dits « contextualisés » (ELMo (Peters *et al.*, 2018), BERT (Devlin *et al.*, 2018)) capables de produire des représentations de mots qui dépendent du contexte. Malgré les gains en performance apportés par ces modèles (cf. classement GLUE), la complexité de leurs architectures implique un coût d'entraînement nettement plus important [1] [2] que pour des approches plus classiques (Word2vec (Mikolov *et al.*, 2013), Glove (Pennington *et al.*, 2014), fastText (Bojanowski *et al.*, 2017)). Ainsi, la tendance générale est d'utiliser les versions *pré-entraînées* plutôt qu'entraîner soi-même ces modèles.

Aujourd'hui, le choix le plus populaire en matière de plongements contextualisés semble être celui du modèle BERT, pour lequel on trouve aussi bien des versions pré-entraînées pour le domaine dit « général » que pour des domaines spécialisés (e.g. BioBERT (Lee *et al.*, 2020) et BlueBERT (Peng *et al.*, 2019) pour le domaine médical, SciBERT (Beltagy *et al.*, 2019) pour le domaine scientifique). Contrairement aux versions générales qui sont intégralement entraînées sur des corpus généraux, ces versions spécialisées semblent toutes être issues d'une procédure standard : partir du modèle BERT entraîné pour le domaine général puis poursuivre l'entraînement de celui-ci sur des textes spécialisés. Cette stratégie apporte des gains en performance incontestables par rapport à l'usage direct d'un modèle général (Alsentzer *et al.*, 2019; Si *et al.*, 2019) mais il semble néanmoins légitime de vouloir comparer ces modèles à des versions entraînées directement sur des textes spécialisés, sans passer par un entraînement en domaine général.

Dans le cadre de ce travail, nous nous concentrons sur le domaine médical pour lequel nous étudions l'impact de trois paramètres sur la performance finale du modèle BERT : le domaine du vocabulaire utilisé (général vs. médical), le corpus d'entraînement initial (général vs. médical vs. mélange des deux) et le corpus de spécialisation (aucun vs. médical). Pour une comparaison plus équitable, nous entraînons nous-mêmes tous les modèles en utilisant exactement les mêmes hyper-paramètres, puis nous évaluons ces modèles sur un ensemble varié de tâches classiques du domaine biomédical : détection de concepts médicaux (i2b2/VA 2010 (Uzuner *et al.*, 2011)), implicature textuelle (MEDNLI (Romanov & Shivade, 2018)) et extraction de relations (ChemProt (Krallinger *et al.*, 2017), DDI (Herrero-Zazo *et al.*, 2013)). Toutes les expériences sont effectuées en langue anglaise.

Nos contributions sont les suivantes :

— nous effectuons une analyse préliminaire sur l'impact du vocabulaire choisi sur la gestion des mots hors vocabulaire par BERT. Nous constatons ainsi une différence notable entre un vocabulaire général et médical pour traiter des termes techniques du domaine médical ;

— nous effectuons une comparaison équitable de plusieurs modèles BERT ayant des degrés de spécialisation différents. Nous observons alors que la stratégie standard consistant à ré-entraîner un modèle général obtient des performances similaires aux modèles entraînés directement sur des corpus médicaux ;

— nous partageons notre code afin de permettre la reproduction de nos résultats, et partageons nos modèles pré-entraînés pour le domaine médical.

Nous commencerons par introduire les principes de BERT (section 2), notamment ceux qui sous-

1. Entraîner un modèle BERT nécessite plusieurs cartes GPU sur une période pouvant atteindre plusieurs semaines.
2. Plusieurs benchmarks ont été effectués par Nvidia et sont consultables ici.

tendent les hypothèses que nous voulons tester (section 3), puis nous présenterons nos expériences (section 4) et leurs résultats (section 5) avant de conclure (section 6).

2 BERT

BERT (Devlin *et al.*, 2018) est un modèle neuronal de plongements lexicaux utilisant une succession de couches Transformer [3] (Vaswani *et al.*, 2017) afin de produire des représentations contextualisées. Ce modèle est entraîné sur deux tâches : une tâche de Modélisation du Langage Masquée (Masked Language Modelling - MLM) et une tâche de Prévision de la Phrase Suivante (Next Sentence Prediction - NSP). Cette section décrit l'architecture de BERT ainsi que la procédure employée pour l'entraîner. Dans tout ce qui suit, nous ferons la distinction entre deux phases :

— la phase de *pré-entraînement*, qui permet au modèle d'apprendre à produire des plongements contextualisés via les deux tâches MLM et NSP ;
— la phase d'*adaptation à une tâche*, où BERT est utilisé comme générateur de plongements au sein d'un modèle plus large qui est intégralement entraîné sur une tâche cible.

2.1 Description du modèle

2.1.1 Segmentation et représentation de l'entrée

Contrairement aux approches classiques à base de mots, BERT utilise un vocabulaire comprenant un mélange de mots et de sous-mots appelés *wordpieces* (Wu *et al.*, 2016). Cela lui permet de remédier au problème des mots hors vocabulaire en découpant chaque mot inconnu en une séquence de sous-mots faisant partie du vocabulaire [4].

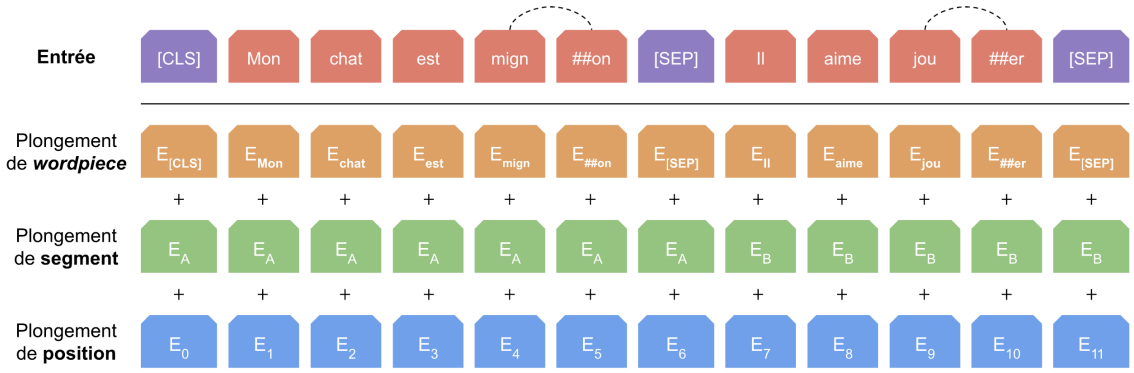


FIGURE 1 – Représentation d'une entrée dans BERT. Les plongements en entrée sont la somme de plongements de *wordpiece* (mot / sous-mot), de plongements de segment (phrase A / phrase B) et de plongements de position (Devlin *et al.*, 2018)

Lors de la phase de *pré-entraînement*, BERT prend systématiquement un couple de phrases en entrée. Ces phrases sont alors segmentées soit en mots, soit en sous-mots, avec un symbole spé-

3. En particulier, il s'agit de la partie « encodeur » de l'architecture du Transformer.
4. Si nécessaire BERT peut descendre jusqu'au niveau du caractère pour décomposer n'importe quel mot en entrée.

cial *[CLS]* en début de séquence et des symboles spéciaux *[SEP]* à la suite de chaque phrase. Chaque élément de l'entrée est alors représenté par un vecteur (*token embedding*) issu d'une matrice de plongement. Puis, afin d'injecter une notion de position et de distinguer plus facilement les éléments issus de chacune des phrases en entrée, on ajoute à ce vecteur initial un plongement de position (*position embedding*) ainsi qu'un plongement de segment (*segment embedding*). La vue complète de ces entrées est représentée en figure 1.

2.1.2 Contextualisation et couches Transformer

BERT transforme chaque paire de phrases en entrée en une séquence de vecteurs selon la procédure décrite précédemment. À ce niveau, la représentation de chaque élément est complètement indépendante de celle des autres. La prochaine étape est alors d'utiliser une série de L couches Transformer appliquant la même transformation [5] (cf. figure 2) afin de produire itérativement des représentations de plus en plus contextualisées.

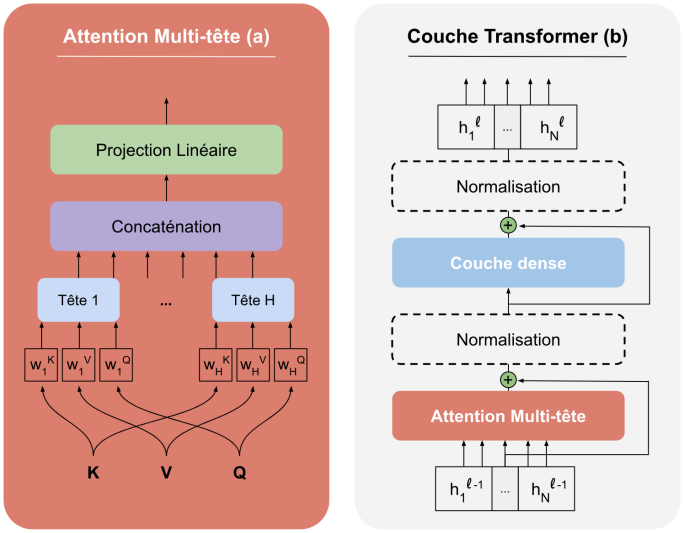


FIGURE 2 – Schéma d'une couche Transformer dans BERT. (a) Au sein du mécanisme d'attention multi-tête, les clés K, valeurs V et requêtes Q sont projetées pour chacune des têtes et servent au calcul d'un produit scalaire d'attention. L'ensemble des produits est ensuite concaténé pour enfin subir une projection linéaire. (b) Au sein de la couche Transformer, la sortie de la couche précédente sert de clé, valeur et requête pour le calcul de l'attention multi-tête

Attention Multi-tête Afin de prendre en compte le contexte global, la couche d'attention multi-tête pondère chaque représentation vis-à-vis du reste des représentations en entrée. Cette pondération dépend de paramètres propres à chaque couche d'attention qui sont ajustés à la tâche au moment de l'entraînement du modèle. De plus, afin de capter des signaux différents, la couche d'attention multi-tête se repose sur plusieurs « têtes » qui calculent à chaque fois une pondération différente des entrées. Enfin, l'ensemble des pondérations est concaténé puis projeté pour produire les représentations en sortie de la couche d'attention.

5. La transformation appliquée par chaque couche est identique mais dépend de paramètres entraînables différents.

2.2 Procédure d'entraînement

Afin d'entraîner BERT à produire des représentations contextualisées utiles pour une large gamme de tâches de TAL, on entraîne celui-ci sur deux tâches : une tâche de modélisation du langage masquée (MLM) et une tâche de prévision de la phrase suivante (NSP).

2.2.1 Modélisation du langage masquée (MLM)

Contrairement aux tâches de modélisation du langage classiques, où l'on cherche à prédire le mot suivant étant donné les mots observés précédemment, BERT est entraîné sur une tâche où l'on masque aléatoirement un mot du texte en entrée, l'objectif étant alors de prédire le mot masqué. Ainsi, grâce à la capacité de l'architecture Transformer à prendre en compte simultanément les contextes droit et gauche du mot cible, cette tâche permet a priori au modèle d'apprendre des représentations encore plus contextualisées que les modèles unidirectionnels tels qu'ELMo [6].

En pratique, les mots cible sont parfois remplacés par un symbole spécial *[MASK]*, parfois remplacés par un autre mot aléatoire et parfois conservés tels quels (cf. figure 3).

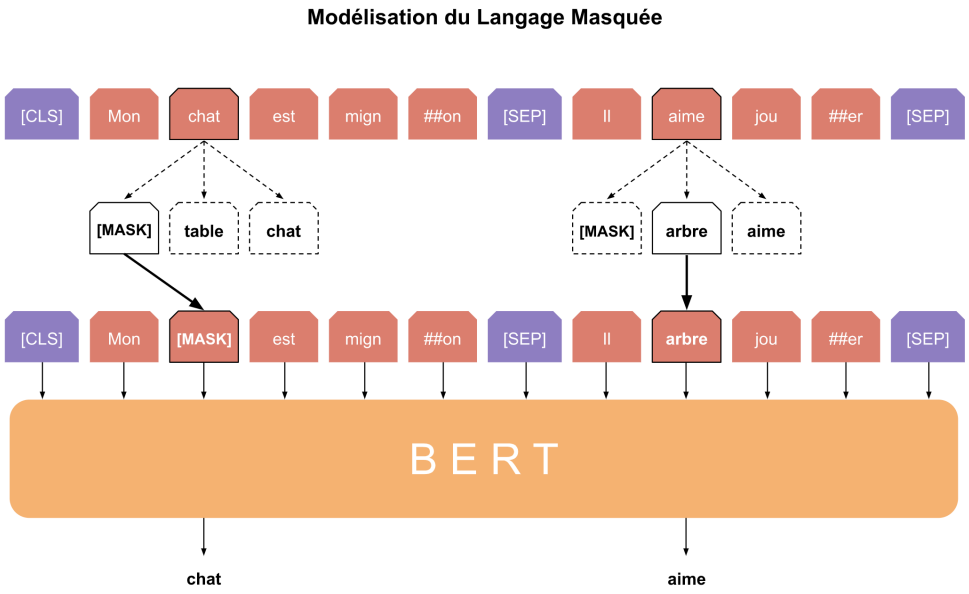


FIGURE 3 – MLM : dans un premier temps, le texte originel est perturbé en modifiant des mots choisis au hasard. Chaque mot est soit remplacé par un symbole spécial *[MASK]*, soit remplacé par un autre mot du vocabulaire ou bien conservé intact

2.2.2 Prévision de la phrase suivante (NSP)

BERT est également entraîné sur une tâche de prévision de la phrase suivante pour laquelle il faut décider si deux phrases en entrée sont consécutives. La justification de cette tâche est d'améliorer

6. Dans ELMo, les plongements sont issus d'une concaténation de représentations unidirectionnelles alors que dans BERT, celles-ci sont naturellement bidirectionnelles.

la performance du modèle sur des tâches où l'objectif est de qualifier la relation entre un couple de phrases (e.g. implicature textuelle). En pratique, la représentation du symbole spécial *[CLS]* est utilisée pour classifier chaque couple de phrases en entrée ainsi que pour toute autre tâche de classification une fois le modèle entraîné.

3 Importance du vocabulaire dans BERT

La segmentation appliquée par BERT se fait en deux étapes : d'abord une segmentation « classique » en mots, puis un découpage en sous-mots (*wordpieces*). Lors de cette seconde étape, BERT découpe autant de fois que nécessaire les mots hors vocabulaire jusqu'à retrouver des *wordpieces* connus. Ainsi, le choix du vocabulaire devrait directement influencer la qualité de cette décomposition, en particulier en domaine médical où le vocabulaire technique est très utilisé.

Pour nous en assurer, nous analysons le résultat de la segmentation d'un corpus médical [7] par deux vocabulaires : l'un du domaine général et l'autre du domaine médical (cf. figure 4). Nous observons alors que le vocabulaire médical a tendance à nettement moins découper les mots que le vocabulaire général, que l'on compte les occurrences dans le texte ou les types de mots distincts.

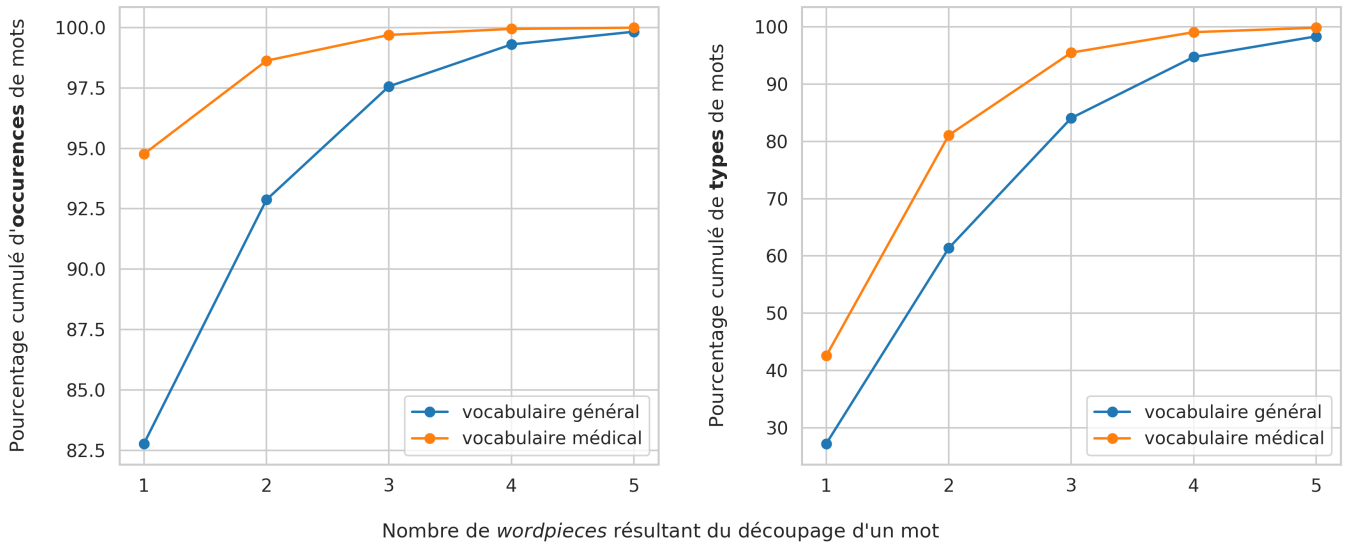


FIGURE 4 – Segmentation de textes médicaux par des vocabulaires de domaines différents

De plus, lorsque nous observons plus qualitativement cette segmentation pour des termes médicaux plus ou moins techniques, nous constatons que la qualité de la segmentation est elle aussi meilleure pour le vocabulaire médical (cf. table 1). En effet, le terme « paracetamol » est directement reconnu par le vocabulaire médical alors qu'il est divisé en *wordpieces* peu porteurs de sens en domaine général. Dans le second cas, « choledocholithiasis » est divisé en deux sous-mots par le vocabulaire médical (« choledoch » et « olithiasis »), tout deux correspondant à des notions médicales alors que le vocabulaire général divise le terme originel en un ensemble de sous-mots là encore peu porteurs de sens. Nous notons cependant que pour des termes plus rares, tels que « borborygmi », le vocabulaire médical semble également être inapte à segmenter le mot en unités porteuses de sens.

7. Il s'agit plus exactement d'un extrait du corpus médical présenté en section 4.1 de l'article.

Terme de référence	Vocabulaire médical	Vocabulaire général
paracetamol	[paracetamol]	[para, ##ce, ##tam, ##ol]
choledocholithiasis	[choledoch, ##olithiasis]	[cho, ##led, ##och, ##oli, ##thi, ##asi, ##s]
borborygmi	[bor, ##bor, ##yg, ##mi]	[bo, ##rb, ##ory, ##gm, ##i]

TABLE 1 – Segmentations en *wordpieces* issues de vocabulaires de domaines différents

4 Expériences

L'approche généralement adoptée est d'entraîner les versions spécialisées de BERT à partir du modèle originel (domaine général) en poursuivant simplement la procédure de pré-entraînement sur des textes spécialisés. Pour évaluer la pertinence de cette stratégie, nous entraînons plusieurs modèles en faisant varier les paramètres suivants : domaine du vocabulaire (général vs. médical), corpus initial (général vs. médical vs. mélange des deux) et corpus de spécialisation (aucun vs. médical).

Dans ce qui suit, nous utilisons l'architecture BERT (base, uncased) qui consiste en $L = 12$ couches Transformer avec pour chacune, $H = 12$ têtes. Nos modèles sont tous appris à partir de textes anglais en minuscule et produisent des plongements contextualisés de dimension 768.

4.1 Différentes configurations de BERT

Corpus	Composition	Nombre de documents	Nombre de mots
Général	Wikipedia (EN)	11 979 758	2 138 764 476
	OpenWebText	3 150 000	1 284 308 223
Médical	MIMIC-III	4 166 225	504 856 155
	PubMed	4 653 528	521 637 990

TABLE 2 – Détail des corpus utilisés pour le pré-entraînement de BERT

Nous notons chaque configuration par un triplet correspondant aux différentes valeurs des paramètres : (V = domaine du vocabulaire, C_1 = corpus initial, C_2 = corpus de spécialisation).

(V = **général,** C_1 = **général,** $\mathrm{C}_2 = \varnothing$) Pour une comparaison équitable, nous entraînons notre propre modèle pour le domaine général. Malgré la redondance que cela représente avec les modèles distribués par (Devlin *et al.*, 2018), entraîner ce modèle nous-même garantit une uniformité des conditions d'entraînement pour tous les modèles comparés. Cependant, nous utilisons le même vocabulaire que le modèle BERT originel : un vocabulaire construit à partir des corpus Wikipédia anglais et BookCorpus (Zhu *et al.*, 2015).

Lors du pré-entraînement, nous utilisons un corpus général (cf. table 2) constitué à partir de Wikipédia anglais ainsi qu'une partie du corpus OpenWebText (Gokaslan & Cohen, 2019) [8]. La portion de ce dernier est choisie de façon à aboutir à une taille de corpus comparable à celle utilisée originellement dans (Devlin *et al.*, 2018).

8. Étant donné que le corpus BookCorpus n'est plus disponible, nous avons remplacé celui-ci par le corpus OpenWebText qui cherche à reproduire WebText, un corpus utilisé pour entraîner le modèle GPT-2 (Radford *et al.*, 2019).

($V =$ **général,** $C_1 =$ **général,** $C_2 =$ **médical**) Nous cherchons ici à reproduire l'approche classique consistant à poursuivre l'entraînement d'un modèle du domaine général sur des textes spécialisés. Plus précisément, tout en gardant le vocabulaire général, nous poursuivons l'entraînement du modèle précédent sur un corpus médical constitué à partir de notes cliniques issues de MIMIC-III (Johnson *et al.*, 2016) et de résumés d'articles scientifiques médicaux issus de PubMed (Fiorini *et al.*, 2018).

($V =$ **médical,** $C_1 =$ **médical,** $C_2 = \varnothing$) Contrairement aux modèles précédents, cette version est directement entraînée sur des textes médicaux. De plus, nous utilisons ici un vocabulaire médical que nous construisons à partir du corpus médical (cf. table 2) en passant par la bibliothèque SentencePiece, qui implémente l'algorithme BPE (Sennrich *et al.*, 2015) [9].

($V =$ **médical,** $C_1 =$ **médical,** $C_2 =$ **médical**) À partir du modèle entraîné directement sur le corpus médical, nous effectuons un second entraînement complet sur ce même corpus afin d'aboutir à une version comparable en terme de durée d'entraînement du modèle ($V =$ général, $C_1 =$ général, $C_2 =$ médical).

($V =$ **médical,** $C_1 =$ **combinaison,** $C_2 = \varnothing$) Il est possible de s'interroger quant à l'intérêt de fusionner deux corpus, l'un général et l'autre médical, afin de pré-entraîner un modèle tel que BERT. En effet, s'il est raisonnable d'entraîner successivement sur les deux corpus, alors il peut être intéressant de considérer le cas où l'on entraîne simultanément sur ceux-ci.

Nous complétons alors notre analyse en entraînant un modèle sur la somme des deux corpus. Afin de garder l'accent sur le domaine médical, nous utilisons ici aussi un vocabulaire médical.

($V =$ **médical,** $C_1 =$ **combinaison,** $C_2 =$ **médical**) Afin d'étudier l'intérêt de partir non pas d'un modèle du domaine général mais d'une version hybride qui aurait vu les deux types de textes, nous poursuivons aussi l'entraînement du modèle précédent sur notre corpus médical.

4.2 Tâches d'évaluation

Nous évaluons nos modèles sur un ensemble varié de tâches biomédicales et cliniques comprenant : détection de concepts médicaux (détection d'entités), implicature textuelle et extraction de relations.

Détection de concepts médicaux Nos modèles sont évalués sur la tâche de détection de concepts médicaux d'i2b2/VA 2010 (Uzuner *et al.*, 2011). Cette tâche consiste en l'extraction de trois types de concepts médicaux : problème (PROBLEM, e.g. « migraine »), traitement (TREATMENT, e.g. « doliprane ») et test médical (TEST, « endoscopie »). Un exemple de note clinique annotée est donné en figure 5.

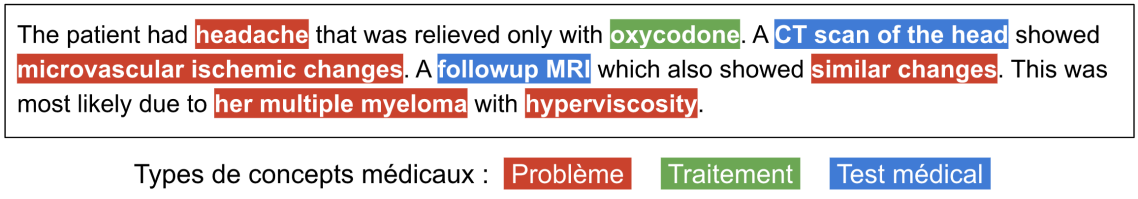


FIGURE 5 – Exemple issu de i2b2/VA 2010. Les entités sont annotées selon le format BIO.

9. Il faut noter que l'algorithme utilisé pour construire le vocabulaire du BERT original n'est pas disponible. Ainsi, pour construire son propre vocabulaire de wordpiece, il faut passer par des algorithmes similaires, tels que le BPE.

Implicature textuelle Nous évaluons également nos modèles sur la tâche d'implicature textuelle MEDNLI (Romanov & Shivade, 2018). Cette tâche consiste à classifier des couples d'extraits de notes cliniques selon 3 catégories : contradiction (CONTRADICTION), implicature (ENTAILMENT) et neutre (NEUTRAL). Des exemples sont donnés en figure 6.

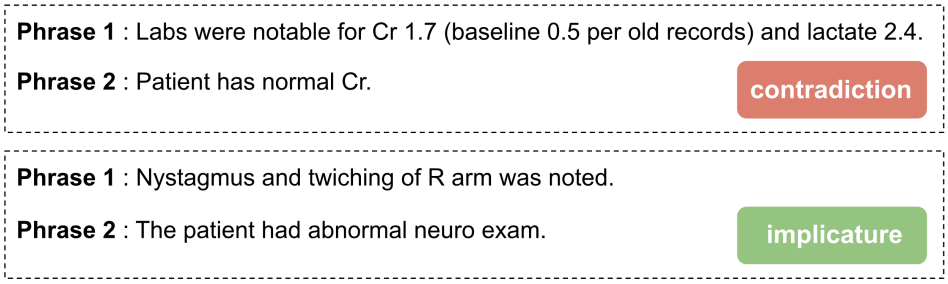


FIGURE 6 – Exemples issus de MEDNLI

Extraction de relations Afin de diversifier nos tâches d'évaluation, nous évaluons nos modèles sur deux tâches d'extraction de relations : ChemProt (Krallinger *et al.*, 2017), issue de la compétition BioCreative VI et DDI (Herrero-Zazo *et al.*, 2013), issue de SemEval 2013 - Tâche 9.2. Pour ChemProt, il s'agit de détecter la présence d'interactions entre composés chimiques et protéines en précisant le type de cette interaction : active (CPR:3), inhibe (CPR:4), agoniste (CPR:5), antagoniste (CPR:6) et substrat (CPR:9). Pour DDI, il s'agit de classifier des extraits de textes selon le type d'interaction entre médicaments qui s'y trouve : conseil (DDI-advise), effet (DDI-effect), mécanisme (DDI-mechanism) et interaction (DDI-int). Un exemple est donné pour chaque tâche en figure 7.

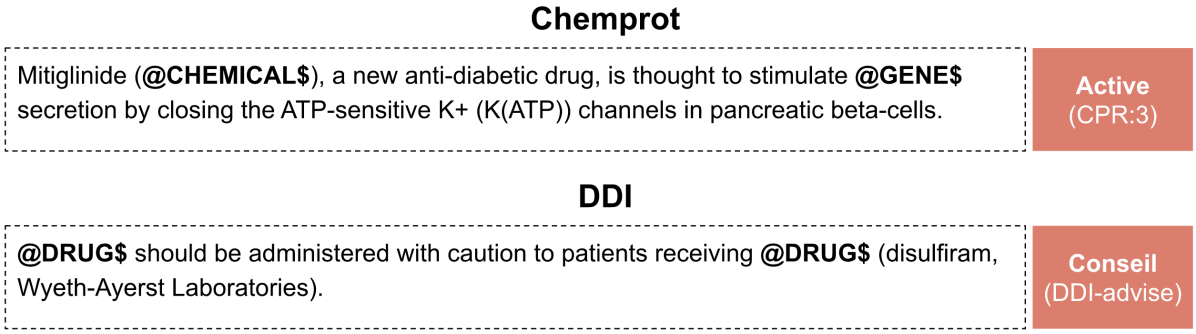


FIGURE 7 – Exemples issus de ChemProt et DDI

Le nombre d'exemples est rapporté pour chaque tâche dans la table 3.

	i2b2/VA 2010	MEDNLI	ChemProt	DDI
Entraînement	24 757	11 232	19 460	18 779
Validation	6 189	1 395	11 820	7 244
Test	45 404	1 422	16 943	5 761

TABLE 3 – Distribution du nombre d'exemples des différentes tâches d'évaluation

4.3 Paramètres des modèles

Afin de faciliter la reproduction de nos résultats, nous partagerons les paramètres utilisés lors du pré-entraînement de nos modèles BERT ainsi que lors de leur évaluation sur nos tâches médicales. De plus, nous mettrons à disposition l'ensemble de nos modèles BERT pré-entraînés ainsi que les codes ayant servi à l'entraînement et à l'évaluation [10].

4.3.1 Paramètres de pré-entraînement

Nous entraînons nos modèles BERT en utilisant 16 cartes graphiques de type Tesla V100-SXM2-16GB. L'implémentation et les paramètres choisis sont ceux fournis dans la base de code de NVIDIA [11]. Chaque entraînement consiste en deux phases :

- **Phase 1** Une série de 3 519 mises à jour (*updates*) sur des paquets (*batch size*) de 8 192 observations de taille 128 avec un taux d'apprentissage (*learning rate*) de 6.10^{-3}. Cette phase dure environ 17 heures.
- **Phase 2** Une série de 782 mises à jour (*updates*) sur des paquets (*batch size*) de 4096 observations de taille 512 avec un taux d'apprentissage (*learning rate*) plus faible valant 4.10^{-3}. Cette phase dure environ 9,5 heures.

L'optimisation a été effectuée via l'algorithme LAMB (You *et al.*, 2019) en employant un taux d'échauffement (*warmup rate*) de $0,01$ et un taux de dégradation (*weight decay*) de $0,01$.

4.3.2 Paramètres d'évaluation

Nous évaluons chaque modèle en effectuant 15 itérations (*epoch*) sur nos données d'entraînement par paquets (*batch size*) de 32 observations. À la suite de chaque itération, nous évaluons le modèle sur un jeu de validation propre à la tâche. Au bout de la dernière itération, le modèle ayant obtenu la meilleure performance sur le jeu de validation parmi les 15 itérations effectuées est retenu.

5 Résultats

Afin de prendre en compte les effets dus aux aspects aléatoires tels que l'initialisation des modèles ou l'échantillonnage des jeux de validation, nous effectuons systématiquement 10 évaluations avec à chaque fois une graine aléatoire (*random seed*) différente. Ainsi, la performance de chaque modèle est calculée en (moyenne$\pm$ écart-type). Les résultats sont présentés en table 4.

5.1 Analyse

Étant donné la complexité de la procédure de pré-entraînement de BERT, il est utile de comparer notre modèle du domaine général au modèle originel : BERT (base). Nous constatons alors que ces deux modèles ont des performances similaires, avec néanmoins un avantage pour BERT (base). Cependant,

10. https://github.com/helboukkouri/recital_2020
11. Plus exactement, nous adaptons ces scripts à nos besoins.

Modèle			Tâche d'évaluation			
V	**C_1**	**C_2**	**i2b2/VA 2010**	**MEDNLI**	**ChemProt**	**DDI**
général	général	∅	85,66 ± 0,18	77,31 ± 0,71	67,47 ± 0,99	75,81 ± 1,02
général	général	médical	89,00 ± 0,17	**84,91** ± 0,46	72,29 ± 0,58	78,82 ± 1,11
médical	médical	∅	88,80 ± 0,10	83,54 ± 0,43	71,30 ± 0,51	79,40 ± 1,15
médical	médical	médical	<u>89,20</u> ± 0,20	84,32 ± 0,73	**72,97** ± 0,46	**80,11** ± 0,79
médical	combinaison	∅	88,32 ± 0,17	82,20 ± 0,79	69,80 ± 0,51	78,90 ± 1,09
médical	combinaison	médical	**89,30** ± 0,11	84,35 ± 0,74	<u>72,80</u> ± 0,87	<u>80,04</u> ± 0,78
BERT (base) (Devlin *et al.*, 2018)			86,42 ± 0,31	77,85 ± 0,63	69,22 ± 0,56	77,89 ± 0,92
BlueBERT (base) (Peng *et al.*, 2019) [a]			88,70 ± 0,21	<u>84,53</u> ± 0,76	68,35 ± 0,61	77,89 ± 0,65

a. Il s'agit ici de la version de BlueBERT entraînée sur PubMed et MIMIC-III.

TABLE 4 – Résultat de l'évaluation des modèles. La performance de i2b2/VA 2010 est calculée en terme de F1 stricte sur les entités à détecter, celle de MEDNLI est calculée en terme de taux d'exemples corrects (*accuracy*) et enfin celles de ChemProt et DDI sont calculées en terme de micro-F1 mesure. La meilleure performance est affichée en gras, la deuxième meilleure est soulignée.

cette différence peut être due aux différences de corpus (Wikipédia et BookCorpus pour BERT (base) et Wikipédia et OpenWebText pour notre version) ou bien aux différences de paramètres de pré-entraînement. Ainsi, étant donné la proximité des performances des deux modèles, nous pouvons considérer notre procédure d'entraînement comme correcte et interpréter le reste des résultats.

Nous nous concentrons dans un premier temps sur les modèles entraînés sur un unique corpus ($C_2 = \varnothing$). Nous vérifions alors une idée intuitive : un modèle BERT entraîné sur un corpus médical avec un vocabulaire médical (V = médical, C_1 = médical, $C_2 = \varnothing$) obtient systématiquement de meilleurs résultats que son équivalent du domaine général. Par ailleurs, nous constatons que la combinaison de corpus (V = médical, C_1 = combinaison, $C_2 = \varnothing$) aboutit à une performance meilleure que celle du modèle général mais moins bonne que celle du modèle médical.

Pour ce qui est des modèles entraînés sur un second corpus, le constat principal est que globalement les performances sont très proches les unes des autres et qu'en tout état de cause, aucune configuration n'est systématiquement meilleure pour toutes les tâches, même celle maximisant le rattachement au domaine médical (V = médical, C_1 = médical, C_2 = médical). Cependant, nous pouvons remarquer de légères différences en faveur des modèles à vocabulaire médical qui semblent être favorisés dans le cas des tâches biomédicales (ChemProt et DDI). En effet, sur les quatre tâches évaluées, i2b2 et MEDNLI font partie du domaine/genre dit « clinique » (textes issus de MIMIC-III) alors que ChemProt et DDI sont du domaine/genre dit « biomédical » (textes issus de PubMed). Étant donné cette catégorisation, nous pouvons observer que l'espace entre le modèle général ré-entraîné sur du médical et le modèle entraîné deux fois sur notre corpus médical est légèrement plus marqué pour les tâches biomédicales, avec notamment une différence moyenne de 1,29 points de F1 sur DDI. Cependant, il nous est impossible de savoir si cet écart est dû aux paramètres étudiés (vocabulaire, corpus) ou bien au type de tâche (les deux tâches biomédicales sont aussi des tâches d'extraction de relations). Par ailleurs, nous notons que la combinaison de corpus n'aboutit jamais à une amélioration importante par rapport au modèle purement médical.

Enfin, nous observons que notre modèle entraîné d'abord sur un corpus général puis sur un corpus médical obtient de meilleures performances que BlueBERT sur les tâches biomédicales, bien que ce dernier soit entraîné dans des conditions similaires.

5.2 Discussion

Étant donné la proximité des performances de certains modèles, il est important de prendre en compte les valeurs moyennes des performances au regard des écarts-types associés. Ainsi, le résultat principal reste tout de même que l'ensemble des modèles atteignent des performances similaires au moment du second entraînement sur le corpus de spécialisation. Par ailleurs, les résultats impliquant notre corpus médical pourraient être amenés à varier en fonction de la taille de celui-ci. En effet, le corpus général est approximativement trois fois plus grand que le corpus médical. Ainsi, lorsque l'on compare des modèles ayant vu l'un ou l'autre, une partie de l'effet observé est sans doute due à la différence de domaine, mais une autre partie pourrait découler de la différence de taille des deux corpus. En particulier, compte tenu de la proximité des performances moyennes des modèles $(\mathrm{V} = \text{général}, \mathrm{C}_1 = \text{général}, \mathrm{C}_2 = \text{médical})$ et $(\mathrm{V} = \text{médical}, \mathrm{C}_1 = \text{médical}, \mathrm{C}_2 = \varnothing)$, il est possible que ce dernier surpasse l'approche classique en utilisant un corpus médical plus conséquent. Enfin, il est utile de préciser que nous avons fait attention à ce que les modèles utilisant une combinaison de corpus soient entraînés aussi longtemps que ceux utilisant un unique corpus. Ainsi, il est possible que ces versions ne soient pas entraînées suffisamment longtemps pour pouvoir observer leur réel potentiel.

6 Conclusion

Dans un contexte où les modèles de plongements « à la BERT » gagnent de plus en plus de terrain, nous avons voulu évaluer le mode d'utilisation standard de ces modèles en domaine spécialisé : ré-entraîner le modèle BERT d'origine sur un corpus spécialisé avant de l'adapter à la tâche d'intérêt. En nous concentrant sur le cas particulier du domaine médical, nous avons comparé plusieurs approches où le modèle initial est entraîné sur un corpus différent (général, médical, général + médical) avant d'être finalement ré-entraîné sur un corpus médical. Nous arrivons alors à la conclusion que, malgré les différences initiales des modèles suite au premier entraînement (médical > médical + général > général), toutes les configurations aboutissent à des performances sensiblement similaires. Nous pouvons alors en déduire, après factorisation des ressources et du temps nécessaires pour l'entraînement de chacun des modèles, que l'approche préférable demeure celle employée par défaut.

Remerciements

Ce travail a été financé par l'Agence Nationale de la Recherche (ANR) dans le cadre du projet ADDICTE (ANR-17-CE23-0001). Nous remercions également Junichi Tsujii ainsi le centre japonais de recherche en intelligence artificielle AIRC [12] pour nous avoir permis d'utiliser le cluster ABCI [13] afin d'effectuer l'ensemble de nos expériences.

12. https://www.airc.aist.go.jp/en/intro/
13. https://abci.ai/

Références

ALSENTZER E., MURPHY J. R., BOAG W., WENG W.-H., JIN D., NAUMANN T. & MCDERMOTT M. (2019). Publicly available clinical bert embeddings. *arXiv preprint arXiv :1904.03323*.

BELTAGY I., LO K. & COHAN A. (2019). SciBERT : A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3606–3611.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

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*.

FIORINI N., LEAMAN R., LIPMAN D. J. & LU Z. (2018). How user intelligence is improving PubMed. *Nature biotechnology*, **36**(10), 937–945.

GOKASLAN A. & COHEN V. (2019). OpenWebText corpus. http://Skylion007.github.io/OpenWebTextCorpus.

HE K., GKIOXARI G., DOLLÁR P. & GIRSHICK R. B. (2017). Mask R-CNN. *CoRR*, **abs/1703.06870**.

HERRERO-ZAZO M., SEGURA-BEDMAR I., MARTÍNEZ P. & DECLERCK T. (2013). The DDI corpus : An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, **46**(5), 914–920.

HOWARD J. & RUDER S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv :1801.06146*.

JOHNSON A. E., POLLARD T. J., SHEN L., LI-WEI H. L., FENG M., GHASSEMI M., MOODY B., SZOLOVITS P., CELI L. A. & MARK R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific data*, **3**, 160035.

KRALLINGER M., RABAL O., AKHONDI S. A. *et al.* (2017). Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, p. 141–146.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

PENG Y., YAN S. & LU Z. (2019). Transfer learning in biomedical natural language processing : An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, p. 58–65.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). GloVe : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543.

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., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). Improving language understanding by generative pre-training. *URL https ://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

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

ROMANOV A. & SHIVADE C. (2018). Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv :1808.06752*.

SENNRICH R., HADDOW B. & BIRCH A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv :1508.07909*.

SI Y., WANG J., XU H. & ROBERTS K. (2019). Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, **26**(11), 1297–1304.

UZUNER Ö., SOUTH B. R., SHEN S. & DUVALL S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, **18**(5), 552–556.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

WANG D. & ZHENG T. F. (2015). Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, p. 1225–1237 : IEEE.

WU Y., SCHUSTER M., CHEN Z., LE Q. V., NOROUZI M., MACHEREY W., KRIKUN M., CAO Y., GAO Q., MACHEREY K. *et al.* (2016). Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*.

YOU Y., LI J., REDDI S., HSEU J., KUMAR S., BHOJANAPALLI S., SONG X., DEMMEL J., KEUTZER K. & HSIEH C.-J. (2019). Large batch optimization for deep learning : Training BERT in 76 minutes. In *International Conference on Learning Representations*.

ZHU Y., KIROS R., ZEMEL R., SALAKHUTDINOV R., URTASUN R., TORRALBA A. & FIDLER S. (2015). Aligning books and movies : Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, p. 19–27.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Évaluation systématique d'une méthode commune de génération

Hugo Boulanger[1]
(1) Université Paris-Saclay, LIMSI, Campus Universitaire bâtiment 507, Rue du Belvédère, 91400 Orsay, France
prénom.nom@limsi.fr

RÉSUMÉ

Avec l'augmentation de l'utilisation du traitement automatique des langues arrivent plusieurs problèmes dont l'absence de données dans les nouveaux domaines. Certaines approches d'apprentissage tel que l'apprentissage zero-shot ou par transfert tentent de résoudre ces problèmes. Une solution idéale serait de générer des données annotées à partir de bases de connaissances des domaines d'intérêt. Le but de notre travail est d'évaluer une méthode de génération simple et de trouver les critères permettant de la mettre en oeuvre correctement. Pour cela, nous comparons les performances d'un modèle obtenu sur des tâches d'annotation quand il est entraîné sur des données réelles ou sur des données générées. Grâce aux résultats obtenus et à des analyses effectuées sur les données, nous avons pu déterminer des bonnes pratiques d'utilisation de cette méthode de génération sur la tâche d'annotation.

ABSTRACT

Systematic evaluation of a common generation method.

As natural language understanding expands, new domains aim to use this tool. This expansion causes many problems, such as the scarcity of data for these domains. Machine learning approaches such as transfer or zero-shot learning deal with the lack of data, but such methods require labeled data. An ideal solution to this problem would be to generate labeled data from in-domain, highly available data such as ontologies. The goal of our work is to evaluate a simple generation method and to find out criterion to make the method best applicable. Systematic evaluation of the generation method is done by generating and augmenting datasets for sequence labelling benchmarks and comparing the performances obtained from those datasets to real datasets. This paper presents the results of such method and analysis of the original datasets to flesh out guidelines on how to properly use generation to obtain good performances on the associated task.

MOTS-CLÉS : TAL, augmentation de données, génération de données synthétiques.

KEYWORDS: NLP, data augmentation, data generation.

1 Introduction

Le traitement automatique de la langue (TAL) est utilisé dans de plus en plus de domaines et se sert de systèmes d'apprentissage pour résoudre les diverses tâches le composant. Cette recrudescence des systèmes d'apprentissage est accompagnée d'une hausse de la demande de données annotées pour différentes tâches et différents domaines. En revanche, la prise de conscience récente sur l'utilisation des données rend le procédé de récolte de données plus difficile. Dans ce contexte, il devient nécessaire de travailler sur des méthodes permettant d'utiliser des données existantes pour résoudre de nouvelles

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 43–56.
HAL : hal-02786185.

tâches ou travailler sur de nouveaux domaines. Les exemples les plus communs sont l'utilisation d'apprentissage par transfert (Pan & Yang, 2010; Ruder, 2019) ou d'augmentation (Kafle *et al.*, 2017). Ces techniques ont cependant besoin de données annotées du domaine. Il serait intéressant de pouvoir s'affranchir d'une telle contrainte en générant des données à partir de données plus facilement accessible tel que des bases de connaissances ou des ontologies.

Dans cet article, nous faisons une analyse systématique d'une méthode commune de génération de données annotées. Cette méthode est la méthode de remplissage de patrons. Le principe de cette méthode est d'utiliser des énoncés à trou, les patrons, pour créer de nouveaux exemples en remplissant les trous par des mentions ayant les mêmes concepts que les trous. Cette méthode est illustrée Figure 1. Nous évaluons la performance de la méthode de génération à travers la performance d'un modèle entraîné sur les données générées. C'est ce qui sera appelé performance, sauf indication contraire. Nous comparons les performances entre les modèles entraînés sur les données de référence et les modèles entraînés sur les données générées à partir des mêmes données de références. A notre connaissance, aucune autre étude n'a effectué d'examen systématique de ce type de méthode de génération pour une tâche d'étiquetage. Notre contribution n'est pas la méthode de génération en elle-même, car elle est déjà largement utilisée, mais c'est la connaissance de l'impact des données générées par ces méthodes sur les performances des modèles d'apprentissage, et c'est aussi la connaissance de la meilleure façon d'utiliser cette méthode de génération.

2 État de l'art

Au sein du TAL, des méthodes de génération ont déjà été utilisées pour créer de nouvelles tâches, comme les tâches bAbI (Weston *et al.*, 2016) dans le contexte des systèmes de dialogue qui visaient à fournir des données pour former des modèles de dialogue de bout en bout. Un examen de la qualité de certaines des tâches bAbI a été effectué dans l'article présentant les hybrid code network (Williams *et al.*, 2017). Il s'avère que la méthode de génération utilisée était trop systématique et que de simples systèmes tels que des systèmes à base de règles ont facilement atteint une performance maximale.

Une approche similaire a été étudiée afin de construire un ensemble de données pour entraîner un modèle de compréhension dans le contexte d'un assistant médical (Neuraz *et al.*, 2018). Dans ce travail, la méthode de génération basée sur le remplissage de patrons et une méthode basée sur la paraphrase ont été comparées. Cette étude montre qu'il est possible d'utiliser des méthodes de génération naïves dans le contexte de l'absence de données. Elle souligne également que l'utilisation de la méthode de paraphrase afin d'augmenter les données n'améliore pas de manière significative les performances dans cette tâche.

Des méthodes plus complexes de génération et d'augmentation existent, comme l'utilisation d'auto-encodeurs variationnels (Kingma & Welling, 2014; Yoo *et al.*, 2019) mais elles nécessitent des données étiquetées, ce qui n'est pas conforme à ce que nous visons. Un autre aspect de notre recherche vise à expliquer les résultats que nous obtenons. À cette fin, les jeux de données de références sont analysées dans le but de trouver des indicateurs qui pourraient être liés aux performances des modèles. Liés à ces analyses sont les analyses effectuées dans (Béchet & Raymond, 2019) décrivant la façon dont les modèles réagissent à chacun des benchmarks et établissant un classement de difficulté des dits benchmarks.

3 Tâche de Compréhension

Enoncé	Is	there	a	flight	to	Atlanta
Etiquettes	O	O	O	O	O	B-city

TABLE 1 – Exemple d'un énoncé et de ses étiquettes. Le but de l'étiquetage est de classer chaque token. Les étiquettes ont deux variantes basées sur leur position suivant le format BIO : B pour le début de la mention, I pour l'intérieur de la mention et O pour extérieur à la mention.

Avant d'expliquer la méthode de génération, il est important de comprendre la tâche que nous voulons résoudre. La tâche abordée dans cet article est la tâche d'étiquetage pour la compréhension du langage naturel. Cette tâche vise à classer chaque token d'un énoncé comme expliqué dans le tableau 1. La classification des énoncés ne fera pas partie de notre analyse.

3.1 Etiquetage

L'étiquetage est une tâche de segmentation et de classification. C'est une tâche pour laquelle des méthodes de génération naïves, telles que le remplissage de patrons, sont déjà largement utilisées. Nous avons utilisé cinq ensembles de données connues pour évaluer le potentiel de cette méthode de génération naïve.

3.2 Benchmarks

Les tâches sur lesquelles la génération est évaluée sont les mêmes que celles utilisés par Béchet et Raymond dans leur article Béchet & Raymond (2019) :

a ATIS (Dahl *et al.*, 1994) ou Air Travel Information System est une tâche de réservation de trajet en avion en Anglais. Exemple : "what airlines go to pittsburgh"

b M2M (Shah *et al.*, 2018) ou Machine Talking To Machines est une tâche de réservation de restaurant et de place de cinéma en Anglais. Exemple : "I would like movie tickets to watch avatar ."

c SNIPS (Coucke *et al.*, 2018) est une tâche d'assistant vocal en Anglais. Example : "Look up Applied Linguistics"

d SNIPS70 est l'ensemble des énoncés utilisés pour l'expérience contenant 70 énoncés par classe d'énoncé dans (Coucke *et al.*, 2018). [1]

e MEDIA (Bonneau-Maynard *et al.*, 2005) est une tâche de réservation de chambre d'hôtel en Français. Ce corpus fût construit à partir de transcriptions orales. Exemple : "euh réserver dans cet hôtel". Dans cet article nous utilisons une version légèrement différente de celle utilisée dans (Béchet & Raymond, 2019) car nous utilisons les tours de parole mixés (plusieurs locuteur parlent dans le même énoncé) et les tours de parole vide (aucun élément n'est étiqueté). [2]

1. SNIPS est composé des énoncés des fichiers "train_class_full.json" et SNIPS70 est composé des énoncés des fichier "train_class.json" de https://github.com/snipsco/nlu-benchmark .
2. Nous nous sommes rendu compte de la différence après les expériences.

Corpus	ATIS	M2M	SNIPS	SNIPS70	MEDIA
# tokens du jeu d'entraînement	50497	45965	121547	15172	94708
Vocabulaire du jeu d'entraînement	867	688	13582	3222	2100
# énoncés du jeu d'entraînement	4478	8148	13284	1600	12916
# énoncés du jeu de développement	500	2116	500	500	1259
# énoncés du jeu de test	893	4800	700	700	3518
# concepts	79	12	39	39	73

TABLE 2 – Description des corpus.

Tous les jeux de données n'ont pas été construits de la même manière, et certains n'ont pas inclus de jeu de développement. Nous avons construit des jeux de développement pour les jeux de données qui n'en avaient pas. MEDIA et M2M ont fourni des jeux de développement. Pour ATIS, nous avons constitué le jeu de développement avec les 500 derniers énoncés du jeu d'entraînement originel car les énoncés ne sont pas rangés par intention. Pour SNIPS et SNIPS70, les jeux de développement que nous avons constitués sont composés des 500 derniers énoncés de leurs jeux d'entraînement respectifs après mélange (les énoncés étaient rangés).

4 Génération

Le but de cet article est de faire une analyse systématique d'une méthode de génération. Pour cela il nous faut une méthode de génération à analyser.

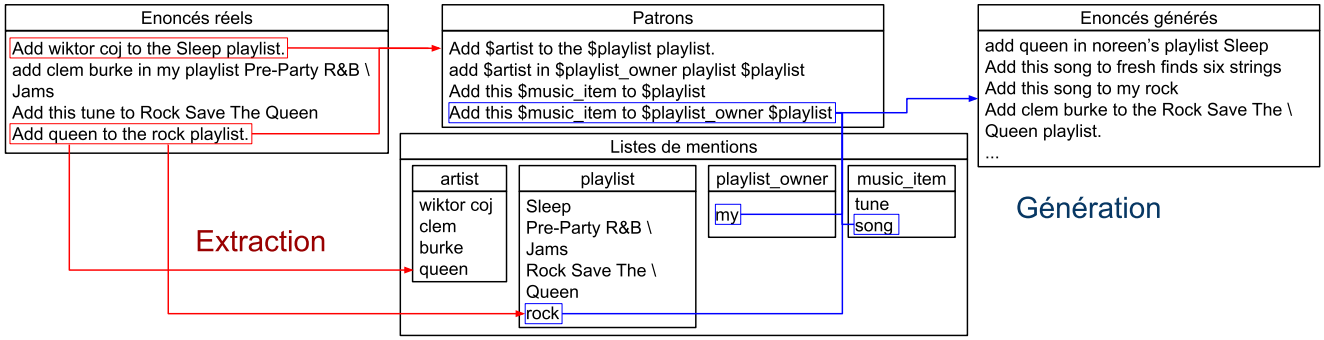


FIGURE 1 – Processus d'extraction et de génération. En rouge, l'extraction effectuée à partir des énoncés de référence afin de constituer nos données brutes pour la génération. En bleu, le processus de génération combine les données brutes afin de créer de nouveaux énoncés. Notre objectif est d'étudier les performances du modèle obtenu à partir des énoncés générés à partir des données brutes extraites des énoncés du benchmark réel.

4.1 Méthode de Génération

La méthode de génération évaluée est une méthode simple de remplissage de patrons, comme le montrent la figure 1 et le tableau 3. Les patrons sont des énoncés dont les mentions ont été remplacées

par leurs classes ou concepts. La méthode de remplissage de patrons consiste à remplir les concepts trouvés dans un patron, $C_{pattern}$, avec les mentions des concepts respectifs, m_c. Cela donne une quantité d'énoncés possibles, p, qui explose très rapidement :

$$p_{pattern} = \prod_{c \in C_{pattern}} Card(m_c)$$

Comme il n'est pas possible de générer tous les exemples pour toutes les quantités de données, le nombre d'énoncés générés a été choisi pour être de $n = 20.000$ pour toutes les différentes quantités de données. Chaque modèle est utilisé pour générer des énoncés $q = n/\#patterns$ par échantillonnage sans remplacement. Cette étape est simplifiée en faisant un échantillonnage des indices se trouvant dans $[\![0, p_{pattern}]\!]$ et en faisant l'extraction des indices des mentions dans leur liste grâce à une série de divisions euclidiennes. Si le modèle ne peut pas générer suffisamment d'énoncés, les énoncés générés sont répétés jusqu'à ce que la quantité requise soit atteinte.

Énoncé	Is	there	a	flight	to	Atlanta
Étiquette	O	O	O	O	O	B-city
Patron	Is	there	a	flight	to	$city
Énoncé généré	Is	there	a	flight	to	Paris

TABLE 3 – Exemple du procédé d'extraction et de génération. Le patron ($3^{ème}$ ligne) est extrait d'un énoncé étiqueté ($1^{ère}$ et $2^{ème}$ lignes). Ce patron est rempli par des mentions ayant le même concept ($city est remplacé par Paris) pour former un nouvel énoncé ($4^{ème}$ ligne). Dans cet exemple, la longueur de la mention est la même, mais en pratique elle peut changer (par exemple : $city peut être rempli par New York).

4.2 Evaluation de la méthode de génération

L'évaluation de la méthode de génération se fait à travers les performances d'un modèle appris sur les données générées, puis validé sur les données de développement et testé sur les données de test. La comparaison entre les performances de ces modèles et celles des modèles entraînés sur les données réelles donne une indication des performances de la génération. Pour que cette comparaison soit pertinente, les modèles et les mentions utilisés pour la génération sont extraits des données réelles, comme le montre la figure 1.

5 Expériences et résultats

Dans cette section, nous décrivons les expériences et les conclusions qui peuvent être tirés des résultats.

5.1 Préparation des données

Nous voulons tester les performances d'un modèle sur plusieurs quantités de données pour connaître l'intérêt de la méthode de génération en fonction des données disponibles. Pour cela nous préparons

les données réelles servant de données brutes. Nous effectuons un découpage en parties de taille croissante contenant les parties plus petites. Cela nous garantit qu'il y a de plus en plus d'information dans les jeux de données. Les données brute servent ensuite à construire les jeux d'entraînement comme décrit Figure 2. Les modèles sont entraînés sur ces données et produisent les résultats obtenus sur les figures 3 et 4 sur les données de test. Nous avons testé différentes manières de distribuer les données au cours de leur génération. Les changements dans les distributions ont été effectués en modifiant la manière dont les patrons et les mentions étaient distribués dans leurs listes respectives, comme le montre la figure 1, puisque notre méthode de génération distribue uniformément les mentions trouvées dans les listes au sein des patrons.

Les multiples distributions testées sont :

Uniforme Les patrons et mentions sont distribué uniformément(c'est à dire qu'il n'y a qu'une apparition dans une liste d'un patron ou d'une mention).

Distribution réelle des mentions Les mentions sont distribuées avec leur fréquences réelle d'apparition (c'est à dire que les listes de mentions contiennent le nombre d'apparition d'une mention). Les patrons sont distribués uniformément.

Distribution réelle de patrons Les patrons sont distribués avec leur fréquences réelle d'apparition et les mentions sont ditribuée uniformément.

Distributions réelles Les patrons et les mentions sont distribuées avec leurs fréquences réelles.

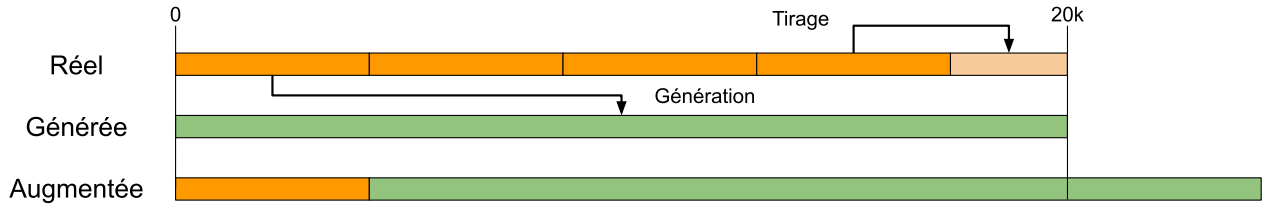


FIGURE 2 – Préparation des données. Les briques oranges symbolisent les données réelles dans leur taille d'origine. Les briques vertes symbolisent les 20.000 générés. Sur la première ligne est représentée la façon dont les données réelles ont été préparées pour l'entraînement. L'échantillonnage est effectué à chaque époque afin de ne pas perturber la distribution sur la totalité de l'entraînement. Sur les deuxième et troisième lignes sont représentées les façons dont les données générées et augmentées ont été préparées.

Pour pouvoir comparer les ensembles générés avec les données réelles utilisées pour les générer, les ensembles de données réels ont été multipliés en longueur afin d'atteindre des énoncés de 20.000, comme le montre la figure 2.

L'augmentation a également été évalué après la génération. Dans ce contexte, la méthode de génération *Distribution réelle des mentions* et la méthode de génération *Distributions réelles* ont été utilisées pour l'augmentation car elles ont donné de meilleurs résultats que les autres méthodes (voir figure 3).

Modèle	ATIS	M2M	SNIPS	SNIPS70	MEDIA
BiLSTM	95.3	91.3	91.8	76.8	84.6
Référence	93.9	92.5	91.8	74.1	85.6

TABLE 4 – Performances du modèle obtenu sur les datasets complets. Le BiLSTM est notre modèle. Le modèle référence est le BiGRU + CRF de (Béchet & Raymond, 2019).

5.2 Système

Le modèle entraîné est un BiLSTM à deux couches [3] (Hochreiter & Schmidhuber, 1997; Greff *et al.*, 2016) de taille caché 128 est ensuite entraîné sur les données. La couche de plongement lexicaux est initialisée avec des vecteurs provenant de Word2Vec (Mikolov *et al.*, 2013) entraîné sur les données d'entraînement. L'ensemble de développement réel est utilisé pour chaque entraînement. La sélection du modèle a été faite en prenant le score de F-mesure le plus élevé obtenu sur l'ensemble de développement. Le score de F-mesure est ensuite calculé sur l'ensemble de test. Le tableau 4 présente l'état de l'art pour les tâches évaluées et les résultats obtenus par notre modèle entraîné sur les données d'entraînement réelles. D'une manière générale, les résultats sont comparables, ce qui rendra nos analyses applicables à tout type de systèmes similaires.

6 Résultats

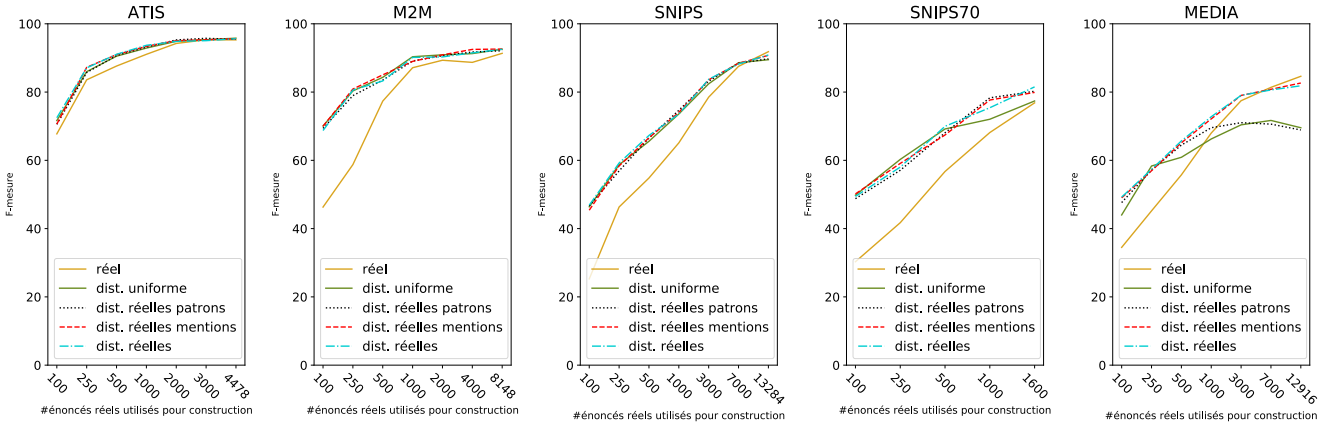


FIGURE 3 – Score de F-mesure obtenus avec conlleval, en mode *génération seule*. Les scores sont obtenus sur le jeu de test à partir de modèles entraînés sur les données préparées à partir du nombre d'énoncés réels trouvés en abscisse.

Le score F-mesure a été mesuré sur l'ensemble des tests de chaque corpus. En mode génération seule, d'après les résultats présentés dans la figure 3, la génération améliore les performances du modèle pour de petites quantités de données, qu'importe la méthode de distribution utilisée. À mesure que les données initiales sont de plus en plus grandes, l'écart entre les modèles entraînés sur les données générées uniformément et les données réelles est réduit, à l'exception de la tâche MEDIA où les modèles entraînés sur les données générées uniformément perdent en performance. Pour la tâche

3. Hyperparamètres : optimiseur Adam avec un taux d'apprentissage initial de 0,001 , plongement lexicaux initialisé avec Word2Vec de taille 300 sur les données d'entraînement, CrossEntropy loss.

SNIPS70, il n'y a pas assez de données disponibles pour atteindre le point où l'écart se referme mais la même tendance peut être observée. En ce qui concerne le mode *augmentation de données* (voir Table 4), on constate que dans l'ensemble les résultats obtenus sont légèrement meilleurs que les résultats dans le mode *génération de données*, notamment sur une grande quantité de données.

Les données générées suivant la distribution *Distribution réelle des patrons* ont tendance à être moins efficaces que les autres jeux de données générés suivant une distribution réelle, en particulier sur la tâche MEDIA où les performances sur de grandes quantités de données sont les pires. En général, les sytèmes appris sur les données générées suivant les *Distribution réelle de mentions* et les *Distributions réelles* tendent à donner de meilleurs résultats mais ont toujours des résultats inférieurs pour SNIPS et MEDIA sur les données générées à partir de beaucoup de données brutes.

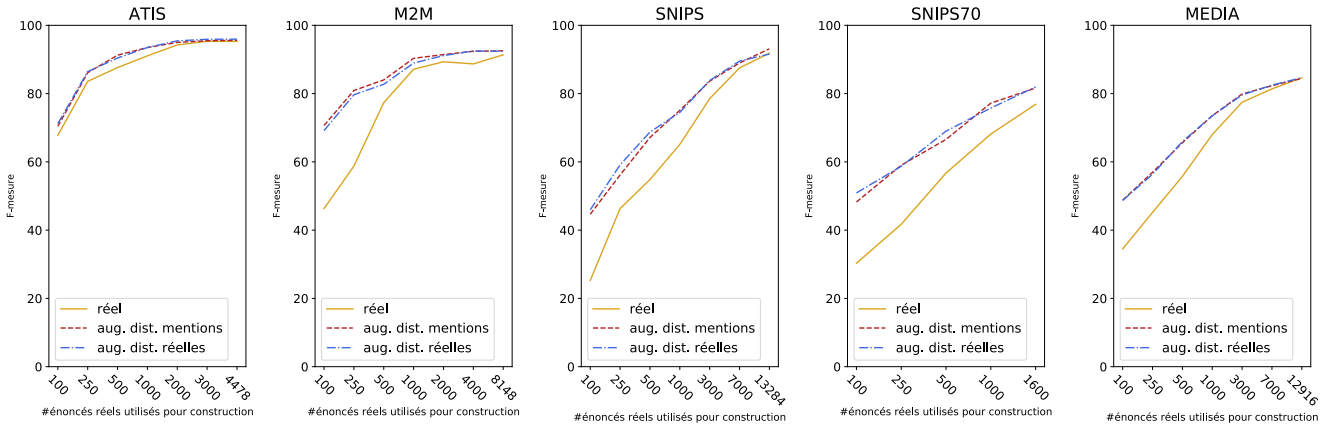


FIGURE 4 – Score de F-mesure des données augmentées calculés avec conlleval, en mode *augmentation de données*. Les scores sont obtenus sur le jeu de test à partir de modèles entraînés sur les données préparées à partir du nombre d'énoncés réels trouvés en abscisse.

7 Analyse des données

Les résultats obtenus à partir des ensembles générés sont bons pour la plupart des tâches. Ce n'est pas le cas pour la tâche MEDIA. Le but de cette section est de trouver les explications de ces performances inférieures à partir d'une série d'analyse de données. Ces analyses visent à découvrir comment le processus de génération perturbe la manière dont les données sont distribuées et comment les distributions initiales des données de certaines tâches peuvent induire leur niveau de difficulté. Un deuxième objectif est de conclure sur de bonnes pratiques concernant la manière dont les données devraient être distribuées pour obtenir de meilleures performances, avec et sans génération.

Comme le montre la figure 3, la modification de la distribution uniforme des mentions en faveur de la distribution réelle a réglé une grande partie des problèmes de performance sur MEDIA. Une analyse plus approfondie est nécessaire pour comprendre la véritable nature de ce changement et pour savoir s'il est possible ou non de trouver des règles générales qui pourraient s'appliquer aux distributions de mentions.

7.1 Distribution des mentions

La distribution des mentions semble avoir une grande importance d'après la figure 3. Ne pas suivre la vraie distribution peut créer des problèmes de performance comme les résultats de MEDIA le montrent. Y a-t-il des indicateurs dans les données de la différence entre les jeux de données avec des mentions uniformément distribuées et les autres jeux de données ? Le premier et le plus simple des indicateurs est la longueur de la mention en nombre de tokens.

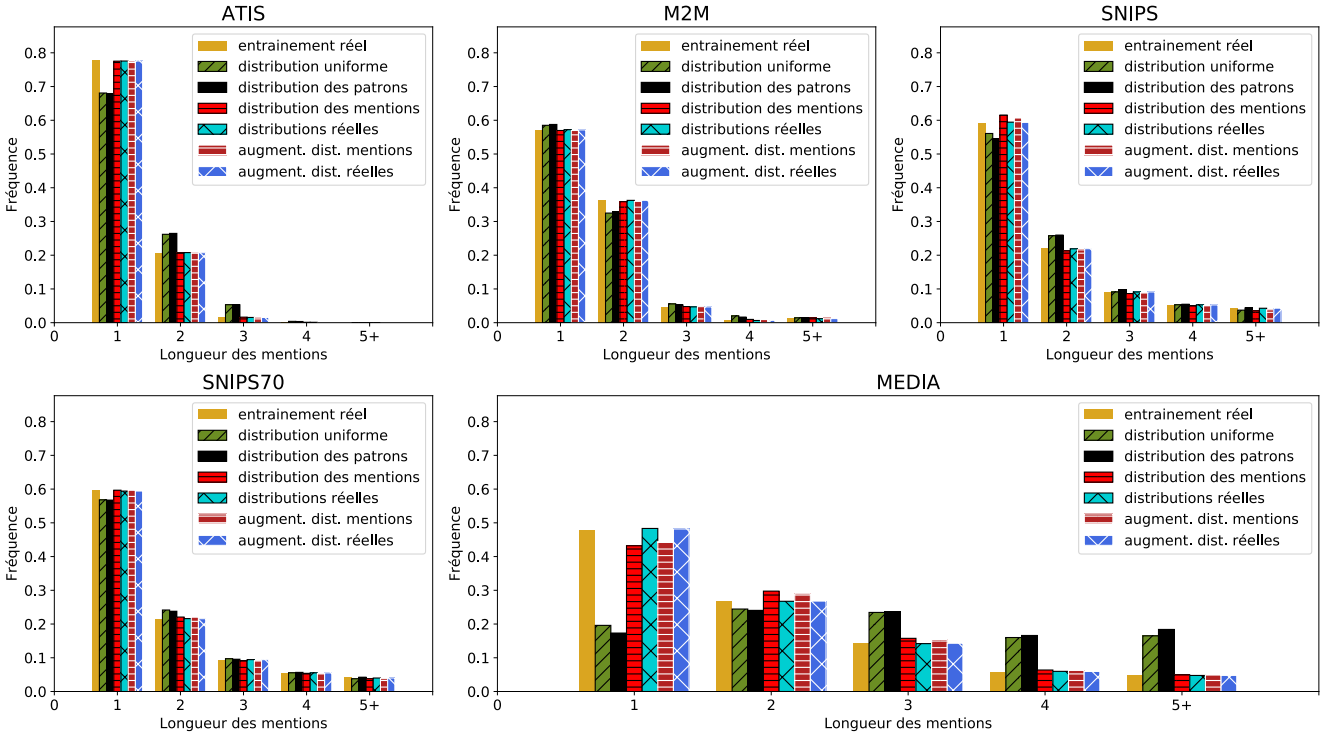


FIGURE 5 – Longueur des mentions en nombre de tokens.

La distribution des mentions basée sur la longueur des mentions, comme le montre la figure 5, montre une tendance commune entre les jeux de données à avoir leurs mentions distribuées de manière similaire à une distribution Zipfienne en fonction de leur nombre de tokens. Pour la plupart des ensembles de données, cette distribution n'est que faiblement influencée par l'échantillonnage uniforme des mentions. Cependant, pour MEDIA, la distribution est complètement faussée comme le montre la figure 5. Cela montre que dans les données réelles de MEDIA, les mentions avec un nombre de tokens inférieur apparaissent avec une plus grande multiplicité que les mentions avec un nombre de tokens supérieur. Ce problème peut être facilement résolu lorsque les mentions sont extraites avec leur multiplicité, mais dans un cas d'une utilisation où il n'y a pas d'énoncés à partir desquels extraire des informations de distribution, cela pourrait constituer un obstacle.

7.2 Impact des patrons

La distribution des patrons ne semble pas avoir d'impact positif sur les performances d'après les résultats Figure 3. Toutefois, cela ne signifie pas que la quantité de patrons n'est pas pertinente pour les performances. L'expérience conçue pour étudier l'impact de la quantité de modèles consiste à étudier l'évolution du score de F-mesure en fonction de la quantité de patrons utilisés. Les listes de mentions

ont été réduites de moitié afin d'avoir un nombre suffisant de mentions pour que le modèle puisse apprendre avec de la variabilité mais leur donner un impact plus faible sur le score. Le pourcentage de mentions de l'ensemble de test trouvé dans l'ensemble d'entraînement, ou recouvrement, a également été calculé afin d'avoir une idée de l'impact des patrons sur les représentations des mentions et de leurs classes.

Les résultats de cette expérience ont montré Figure 6 que le recouvrement des mentions tend à atteindre son plateau final à environ 100 patrons. Cependant, l'augmentation des performances des modèles a tendance à ralentir après 250 à 500 patrons. Cela montre qu'il faut une certaine variété dans le contexte des mentions pour atteindre de meilleures performances.

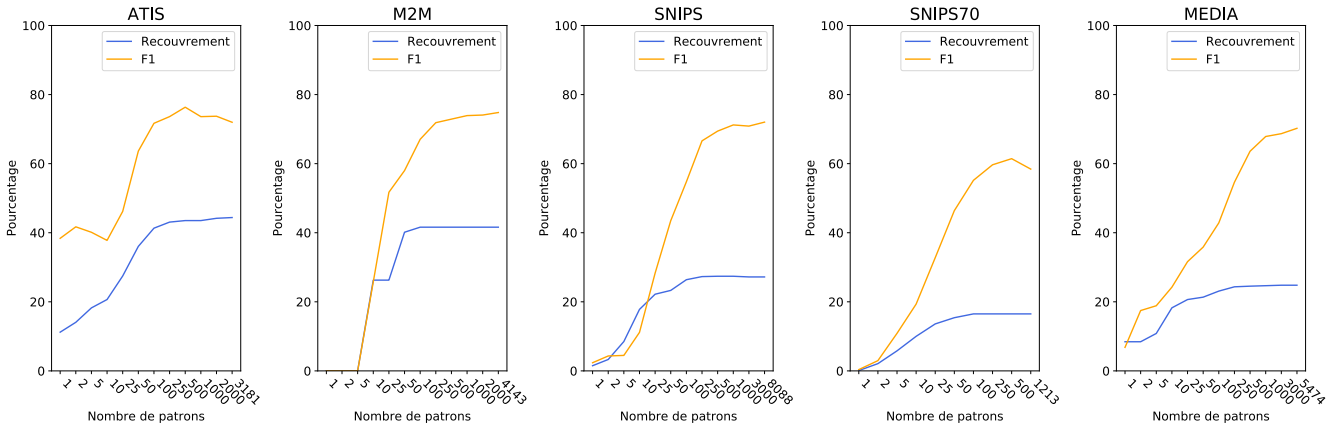


FIGURE 6 – F-mesure dépendant du nombre de patrons. Le recouvrement est le pourcentage de mentions du jeu de test trouvé dans le jeu d'entraînement généré. Cette expérience à été réalisée avec la moitiée des mentions. Les cinq premiers patrons tirés de M2M se trouvent être des patrons vides(ne contienne pas de mentions), ce qui explique les résultats nuls.

7.3 Ambiguïté

Un problème courant dans le traitement du langage naturel est l'ambiguïté. Nous avons choisi de mesurer l'ambiguïté parce qu'elle peut influencer la difficulté d'un jeu de données. Nous mesurons l'ambiguïté en comptant le nombre de concepts dont un token fait partie. Les extérieurs (étiquette O) ne sont pas considérés comme des concepts. L'ambiguïté est faible pour M2M car près de 75% des tokens n'ont pas de concept. SNIPS est peu ambigu, mais sur un nombre élevé de tokens, près de 80% d'entre eux ont un concept, mais peuvent également être extérieurs. ATIS et MEDIA sont ambigus sur une plus petite proportion de leurs tokens, mais avec un nombre plus élevé de concepts par token dans l'ensemble. L'ambiguïté telle que nous la mesurons ne semble pas avoir un rôle significatif dans l'évaluation de la difficulté de la tâche car elle est difficilement comparable pour tous les ensembles de données.

8 Analyses des résultats

Dans cette section, nous compilons et tirons des conclusions sur les résultats et les analyses effectuées. Ces conclusions prennent la forme de lignes directrices sur la manière d'utiliser la méthode de

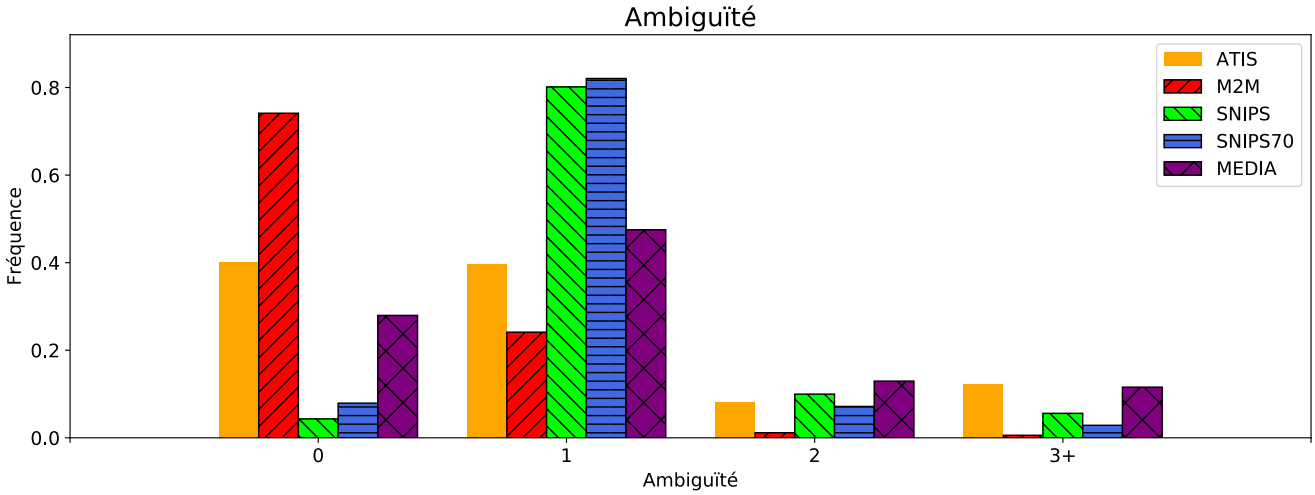


FIGURE 7 – L'ambiguité mesuré comme le nombre de concepts auquel peut appartenir un token. 0 est quand un token ne peut être qu'extérieur. n est quand un token fait partie de n concepts, mais peut aussi être un extérieur. L'ambiguité montré est l'ambiguité du jeu de données réelle de chaque corpus, car l'ambiguité ne varie quasiment pas avec la génération.

génération. Ces lignes directrices sont valables lorsque le modèle d'étiquetage utilisé est un BiLSTM, des recherches supplémentaires sont nécessaires pour pouvoir affirmer que ces lignes directrices réagissent de manière similaire avec d'autres types de modèles (par exemple les CRF).

8.1 Génération

L'utilisation la plus élémentaire d'une méthode de génération est de l'utiliser uniquement pour la génération. Dans cette configuration, nous avons vu qu'il y a quelques points qui doivent être respectés pour générer efficacement. Tout d'abord, nous avons vu que la diversité des patrons est très importante jusqu'à un certain point, après quoi la diminution du rendement peut rendre prohibitif la recherche de nouveaux patrons. Ce point se situe entre 250 et 500 patrons. Le deuxième point principal qui semble ressortir de l'analyse des données est que les mentions doivent être distribuées avec une distribution de type Zipfienne basée sur la longueur des mentions. Le respect de ces points devrait permettre d'obtenir un système aux performances correctes.

8.2 Augmentation

L'utilisation la plus courante de la méthode de génération indiquée ci-dessus est très probablement l'augmentation. Pour de faibles quantités de données, d'après nos observations, l'augmentation et la génération ont tendance à avoir des performances similaires. Il s'agit probablement d'un biais dans la façon dont les données générées et les données réelles sont distribuées dans les ensembles de données augmentées. Si on dispose de données réelles, il est possible d'utiliser les distributions des données réelles pour la génération ou l'augmentation. Cela améliore globalement les résultats. La meilleure utilisation de l'augmentation se situe au niveau des grandes quantités de données où elle tend à donner de meilleurs résultats que les données réelles ou générées, ce qui remet en question la répartition des données notre augmentation.

8.3 Classement des Corpus

Les analyses effectuées sont complémentaires à ce qui est décrit dans l'article (Béchet & Raymond, 2019). Leur travail est davantage axé sur les modèles et les performances, contrairement à notre approche, qui est davantage axée sur le contenu des corpus. Les critères que nous utilisons pour classer les tâches sont : la performance de base de la méthode de génération, la longueur des mentions et le taux d'ambiguïté. Dans l'ensemble, le classement reste le même que celui présenté dans (Béchet & Raymond, 2019) : M2M > ATIS > SNIPS > SNIPS70 > MEDIA.

9 Conclusions et travaux futurs

Nous avons constaté et validé au cours de ce travail que la méthode de génération par remplissage de patrons est utile pour de faibles quantités de données. Cependant, il n'y a que peu d'amélioration significative des performances pour des quantités plus importantes de données si cette méthode est utilisée sans aucune information sur la distribution. Les performances peuvent même se dégrader comme le montre MEDIA.

Ces problèmes nous ont poussés à tester d'autres façons d'utiliser la méthode de génération, comme le test avec les distributions de mention réelle ou l'augmentation des données. Cela nous a également poussés à faire une analyse des corpus afin d'essayer de comprendre pourquoi de tels problèmes pouvaient survenir. Nous avons pu constater que les mentions et la manière dont elles sont distribuées dans un ensemble de données est un facteur clé dans les performances obtenues à partir de cet ensemble de données. C'était le principal facteur expliquant pourquoi les performances de MEDIA étaient aussi dégradées qu'elles l'étaient, et en réglant ce problème avec la distribution réelle, nous avons montré que nous pouvons obtenir des performances proches de la réalité avec les données générées. L'analyse que nous avons effectuée confirme également la classification présentée (Béchet & Raymond, 2019).

Nous ne pouvons pas conclure sur une méthode d'utilisation de la génération, mais nous pouvons mettre en évidence les meilleures pratiques. Nos résultats dépendent des modèles que nous avons utilisés pour l'évaluation des performances (ici, un BiLSTM), et les expériences nécessiteraient d'être réalisées avec d'autres types de modèles (par exemple des CRF). Ces meilleures pratiques consistent à mettre l'accent sur la distribution de mentions plus courtes et à disposer d'un nombre raisonnable de patrons pour avoir un contexte varié.

Dans les travaux futurs, nous aborderons les points non résolus tels que la recherche et l'évaluation de méthodes d'estimation des distributions de mentions. Avec les résultats sur l'impact de la variété des patrons, nous travaillerons également sur la conception et l'essai de méthodes de génération ou d'augmentation des patrons. L'augmentation doit également être étudiée plus en détail, en particulier pour les faibles quantités de données pour lesquelles notre préparation actuelle des données pourrait limiter les performances de l'augmentation aux performances du dispositif de génération. En voyant comment la singularité de MEDIA nous a donné des indices pour trouver de meilleures méthodes de génération, les travaux futurs porteront sur d'autres corpus connus d'étiquetage de phrases. Enfin, notre travail ne concernait que la génération des données d'entraînement, mais pour qu'une méthode de génération puisse être utilisée, il est nécessaire d'étudier comment générer au mieux les données d'entraînement et de validation.

Références

BÉCHET F. & RAYMOND C. (2019). Benchmarking Benchmarks : Introducing New Automatic Indicators for Benchmarking Spoken Language Understanding Corpora. In G. KUBIN & Z. KACIC, Éds., *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, p. 4145–4149 : ISCA.

BONNEAU-MAYNARD H., ROSSET S., AYACHE C., KUHN A. & MOSTEFA D. (2005). Semantic annotation of the french media dialog corpus. In *Ninth European Conference on Speech Communication and Technology*.

COUCKE A., SAADE A., BALL A., BLUCHE T., CAULIER A., LEROY D., DOUMOURO C., GISSELBRECHT T., CALTAGIRONE F., LAVRIL T. *et al.* (2018). Snips voice platform : an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv :1805.10190*.

DAHL D. A., BATES M., BROWN M., FISHER W., HUNICKE-SMITH K., PALLETT D., PAO C., RUDNICKY A. & SHRIBERG E. (1994). Expanding the scope of the ATIS task : The ATIS-3 corpus. In *Proceedings of the workshop on Human Language Technology*, p. 43–48 : Association for Computational Linguistics.

GREFF K., SRIVASTAVA R. K., KOUTNÍK J., STEUNEBRINK B. R. & SCHMIDHUBER J. (2016). LSTM : A search space odyssey. *IEEE transactions on neural networks and learning systems*, **28**(10), 2222–2232.

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

KAFLE K., YOUSEFHUSSIEN M. & KANAN C. (2017). Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, p. 198–202.

KINGMA D. P. & WELLING M. (2014). Auto-Encoding Variational Bayes.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, p. 3111–3119.

NEURAZ A., LLANOS L. C., BURGUN A. & ROSSET S. (2018). Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. *arXiv preprint arXiv :1811.09417*.

PAN S. J. & YANG Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, **22**(10), 1345–1359.

RUDER S. (2019). *Neural Transfer Learning for Natural Language Processing*. Thèse de doctorat, National University of Ireland, Galway.

SHAH P., HAKKANI-TÜR D., TÜR G., RASTOGI A., BAPNA A., NAYAK N. & HECK L. (2018). Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv :1801.04871*.

WESTON J., BORDES A., CHOPRA S. & MIKOLOV T. (2016). Towards ai-complete question answering : A set of prerequisite toy tasks. In Y. BENGIO & Y. LECUN, Éds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

WILLIAMS J. D., ASADI K. & ZWEIG G. (2017). Hybrid code networks : practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 665–677, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1062.

YOO K. M., SHIN Y. & LEE S.-G. (2019). Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, p. 7402–7409.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Analyse de la régulation de la longueur dans un système neuronal de compression de phrase : une étude du modèle LenInit

François Buet
Université Paris-Saclay, CNRS, LIMSI,
Campus universitaire bât 508, Rue John von Neumann, F - 91405 Orsay cedex
buet@limsi.fr

RÉSUMÉ

La simplification de phrase vise à réduire la complexité d'une phrase tout en retenant son sens initial et sa grammaticalité. En pratique, il est souvent attendu que la phrase produite soit plus courte que la phrase d'origine, et les modèles qui intègrent un contrôle explicite de la longueur de sortie revêtent un intérêt particulier. Dans la continuité de la littérature dédiée à la compréhension du comportement des systèmes neuronaux, nous examinons dans cet article les mécanismes de régulation de longueur d'un encodeur-décodeur RNN appliqué à la compression de phrase, en étudiant spécifiquement le cas du modèle LenInit. Notre analyse met en évidence la coexistence de deux influences distinctes au cours du décodage : celle du contrôle explicite de la longueur, et celle du modèle de langue du décodeur.

ABSTRACT

Investigating Length Regulation in a Sentence Compression Neural System : a Study on the LenInit Model

Sentence simplification aims to reduce the complexity of a sentence, while retaining its original meaning and fluency. In practice, the target sentence is expected to be shorter than the source, and methods that incorporate an explicit control over output length are particularly relevant. In the wake of litterature which seeks to understand the internal behavior of neural systems, we investigate in this article the length regulation mechanisms for a RNN encoder-decoder applied to sentence compression, specifically studying the case of the LenInit model. Our analysis highlights the coexistence of two distinct influences during the decoding, from the explicit control and from the decoder language model.

MOTS-CLÉS : compression de phrase, seq2seq, longueur, contrôle explicite, probing.

KEYWORDS: sentence compression, seq2seq, length, explicit control, probing.

1 Introduction

La *simplification de phrase* peut être définie comme la tâche qui consiste à réduire la complexité d'une phrase, tout en préservant sa grammaticalité et son sens initial. La notion de complexité peut ici faire référence à plusieurs caractéristiques de la phrase (par exemple le vocabulaire, la syntaxe, les connaissances extérieures pré-supposées), et doit être considérée relativement à l'objectif de l'application. La littérature sur la simplification a développé de nombreuses approches, qui se

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 57–70.
HAL : hal-02786187.

distinguent notamment par le type d'opérations autorisées pour la transformation de la phrase : la simplification *extractive* repose sur la suppression de mot (Knight & Marcu, 2002) ; d'autres méthodes utilisent la *paraphrase*, en accentuant les simplifications lexicales (Horn *et al.*, 2014) ou syntaxiques (Cohn & Lapata, 2008; Specia, 2010; Rush *et al.*, 2015; Takase & Okazaki, 2019).

Souvent, il est attendu que la phrase simplifiée soit plus courte que l'entrée - la tâche devient alors la *compression de phrase*. Dans bien des cas, il est intéressant que le système soit capable de contrôler la réduction de la longueur. En effet, si le but final est d'améliorer la lisibilité d'un texte pour un groupe d'utilisateurs, un contrôle lâche de la longueur est un moyen simple d'adapter le niveau de lecture. Dans le cadre du sous-titrage automatique, un contrôle plus strict est nécessaire pour respecter les contraintes de temps de lecture et de largeur du moniteur (Aziz *et al.*, 2012).

Kikuchi *et al.* (2016) ont proposé *LenInit* [1], une approche pour la compression à base d'encodeur-décodeur, qui contrôle la longueur de la phrase engendrée en introduisant dans l'état initial du décodeur un vecteur qui encode la valeur visée. Afin d'avoir une meilleure compréhension des mécanismes possibles pour la compression de phrase, et dans la continuité des travaux de *sondage* (*probing*) des modèles neuronaux (Shi *et al.*, 2016; Adi *et al.*, 2016; Conneau *et al.*, 2018), nous avons décidé d'analyser la méthode LenInit, en tentant d'estimer ses limites, et en essayant de comprendre les changements qu'elle implique par rapport au fonctionnement interne d'un RNN encodeur-décodeur classique.

Pour cette étude, nous avons créé un corpus artificiel de compression de phrase, et l'avons utilisé pour entraîner une ré-implémentation au niveau caractère de LenInit. Puis nous avons mené trois groupes d'expériences. Premièrement, nous avons effectué des mesures sur la précision du contrôle de longueur par LenInit, et sur la qualité des phrases produites. Deuxièmement, nous avons entraîné un classificateur pour prédire la longueur future (i.e. le nombre de caractères à engendrer avant *fin-de-phrase*) à partir d'un état caché du décodeur, de manière à suivre l'évolution de la représentation de la longueur au cours du décodage. Troisièmement, nous avons tracé l'évolution de la probabilité associée par le modèle aux caractères associés à la fin de phrase. Notre analyse montre que LenInit exerce un contrôle probabiliste sur la longueur, et suggère la coexistence de deux influences distinctes au cours du décodage : celle de l'objectif explicite, et celle du modèle de langue du décodeur.

2 Contexte

Nous décrivons ici le modèle et les données que nous avons utilisés pour nos expériences de compression de phrase au niveau des caractères.

2.1 Réimplementation de LenInit

Le cadre des systèmes *encodeur-décodeur* avec *réseaux de neurones récurrents* (*RNN*) présente un processus en deux phases pour la *transduction* de phrases. D'abord, le côté encodeur reçoit une *phrase d'entrée* x (de longueur l_x) et produit une séquence *d'états cachés* $(h_1, \ldots, h_{l_x})$. Puis, le côté décodeur reçoit les états cachés de l'encodeur et engendre une *phrase de sortie* $\hat{y}$ (de longueur $l_{\hat{y}}$). À chaque étape j de cette seconde phase, le décodeur calcule un *vecteur de contexte* c_j comme une

1. Code disponible à l'adresse https://github.com/kiyukuta/lencon.

combinaison pondérée et normalisée des états cachés de l'encodeur (ce qui est désigné par *mécanisme d'attention*) et l'utilise pour la mise à jour de son propre état caché courant s_j :

$$s_j = f(s_{j-1}, \hat{y}_{j-1}, c_j), \tag{1}$$

où $\hat{y}_{j-1}$ est le mot engendré à l'étape précédente, et f est une fonction non-linéaire. s_j est à son tour utilisé pour calculer la distribution de probabilité associée au prochain mot :

$$p_{decoder}(\hat{y}_j|\hat{y}_{[1;j-1]}, x) = \text{softmax}(g(s_j, \hat{y}_{j-1}, c_j)), \tag{2}$$

où g est une fonction non-linéaire.

Développée par Kikuchi *et al.* (2016), la méthode *LenInit* est fondée sur ce cadre. Nous l'avons réimplémentée, en l'adaptant pour la transduction au niveau des caractères.

LenInit encode la longueur de sortie voulue en multipliant sa valeur scalaire l avec un paramètre appris, le vecteur de longueur V. Cet encodage continu est ensuite introduit au sein du premier état caché du décodeur s_0, comme suit :

$$s_0 = \tanh\left(W_{init}\left[\frac{\sum_{i=1}^{l_x} h_i}{l_x}; L_{param}\right]\right), \quad L_{param} = l \times V, \tag{3}$$

où W_{init} est un paramètre appris. Pendant la période d'entraînement, l est égal à la longueur de la séquence cible de référence, l_y. Pendant la période de test, l est fixé par l'utilisateur : dans nos expériences, $l = r \times l_x$, avec r le *taux de compression* visé.

Nous avons aussi testé des variantes de LenInit, désignées comme LenInit2 et LenInit3, qui se distinguent par la façon de définir L_{param} dans l'équation (3) :

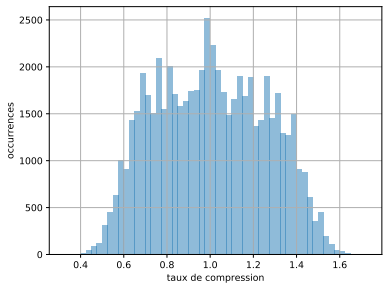
$$\text{LenInit2 :} \quad L_{param} = [l \times V; L_{plong}(l)], \quad \text{LenInit3 :} \quad L_{param} = L_{plong}(l), \tag{4}$$

où $L_{plong}(l)$ est le *plongement* associé à l par une table apprise. L'encodage par la norme de LenInit a en théorie l'avantage de pouvoir marcher pour n'importe quelle longueur, alors qu'un plongement classique (qui apprend indépendamment un vecteur pour chaque valeur) est limité par la fréquence dans les données d'entraînement de la longueur considérée. Avec ces variantes nous avons voulu vérifier le bénéfice lié à un encodage continu, et voir s'il vient au coût d'une perte sur la précision du contrôle de longueur.

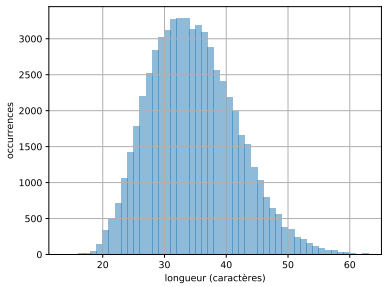
2.2 Un corpus artificiel pour la compression de phrase

Nous avons choisi de mener nos essais sur une tâche plus simple et plus contrôlée que la compression de phrase classique. Pour cela, nous avons créé des données artificielles à l'aide d'un système élémentaire de transduction, qui compresse ou décompresse une phrase source de façon extractive, respectivement en supprimant ou en répétant une partie des caractères. Afin de rendre l'apprentissage d'une telle transformation moins triviale, la décision de supprimer (resp. répéter) un caractère source x_i suit une probabilité p_{sup} (resp. p_{rep}) qui dépend de son « entropie » (autrement dit, qui dépend de sa propension à pouvoir être prédit étant donné le contexte antérieur) :

$$\begin{aligned} p_{sup}(x_i|x_{[i-n+1;i-1]};\theta) &= \max(1, \alpha \times p_\theta(x_i|x_{[i-n+1;i-1]})), \\ p_{rep}(x_i|x_{[i-n+1;i-1]};\theta) &= \max(1, \beta \times (1 - p_\theta(x_i|x_{[i-n+1;i-1]}))), \end{aligned} \tag{5}$$

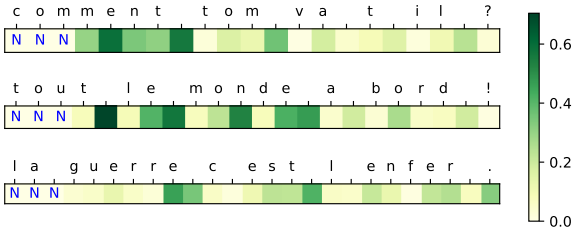


(a) Taux de compression



(b) Longueur source

FIGURE 1: Histogrammes du taux de compression et de la longueur source dans l'ensemble d'entraînement du corpus artificiel.



(a) phrases initiales (sources)

$r = 0,8$	`commttomva t il ?`
$r = 1,2$	`toutt le mmondde a borrd !`
$r = 0,6$	`la guerrc est l enfer`

(b) phrases modifiées (cibles)

FIGURE 2: Exemples de paires dans le corpus. Les phrases sources sont accompagnées d'une carte thermique indiquant pour chaque caractère (excepté les 3 premiers) la probabilité attribuée par un modèle de langue 4-gramme. Les phrases cibles sont compressées ou décompressées, selon r.

où θ est un *modèle de langue* n-gramme au niveau caractère, $p_\theta(x_i|x_{[i-n+1;i-1]})$ est la probabilité d'après p_θ de x_i sachant le contexte des $n-1$ caractères précédents, et α et β sont des facteurs d'échelle (permettant de contrôler la valeur moyenne). Ainsi, une compression devrait supprimer les caractères les plus prévisibles, et une décompression devrait répéter les moins prévisibles.

Notre ensemble de données est constitué de 75 569 phrases de 5 à 10 mots en français, normalisées [2], provenant de Tatoeba [3]. Les phrases cibles sont un mélange de phrases compressées et décompressées. Nous avons défini les valeurs des coefficients α et β de manière à ce que le taux de compression l_y/l_x soit uniformément distribué dans $[0,5; 1,5]$, quand échantillonné sur le corpus entier (Figure 1a) :

$$\alpha = \frac{1-r}{\tilde{p}_\theta}, \quad \beta = \frac{r-1}{1-\tilde{p}_\theta}, \tag{6}$$

où r est une variable échantillonnée uniformément dans $[0,5; 1,5]$ pour chaque phrase (si $r > 1$ une décompression est opérée, sinon une compression), et $\tilde{p}_\theta$ est la probabilité moyenne attribuée par le modèle n-gramme θ aux caractères de référence sur ses données d'apprentissage. θ avait été appris au préalable sur le côté source de l'ensemble d'entraînement.

La figure 1b montre la répartition des longueurs sources de l'ensemble d'entraînement, et la figure 2 présente des exemples de phrases compressées ou décompressées de l'ensemble de développement.

2. Nous avons retiré les diacritiques, majuscules, apostrophes, tirets et virgules.
3. Tatoeba est une base de données de traduction multilingue, constituée de contributions volontaires. `https://tatoeba.org`, diffusé sous licence CC-BY 2.0 FR.

3 Expériences

3.1 Évaluation de LenInit

Précision du contrôle de longueur

Nous nous sommes attachés dans notre premier groupe d'expériences à mesurer la précision avec laquelle LenInit contrôle la longueur de la phrase engendrée. Pour cela nous avons choisi de calculer l'*erreur absolue moyenne* (EAM) et la *racine de l'erreur quadratique moyenne* (REQM) des taux de compression obtenus par rapport aux taux de compression visés, selon les formules suivantes :

$$\text{EAM} = \frac{1}{n}\sum_{i=1}^{n}|\hat{r}_i - r_i|, \quad \text{REQM} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{r}_i - r_i)^2}, \tag{7}$$

où $n = 7457$ est le nombre d'instances dans l'ensemble de développement, et $\hat{r}_i$ et r_i sont respectivement le taux de compression obtenu et le taux de compression visé pour la i-ème phrase transduite.

L'*erreur absolue* (EA) $|\hat{r} - r|$ peut aussi être vue comme la différence entre la longueur produite et la longueur visée $|l_{\hat{y}} - r \times l_x|$ rapportée à la longueur source l_x. Pour compléter nos métriques, nous avons évalué la proportion d'instances pour lesquelles l'erreur absolue est inférieure à 10%, 5%, ou bien est nulle [4] (Section 5.1).

Validité des phrases engendrées

Quoique davantage intéressés par l'aptitude de LenInit à contrôler la longueur, nous avons mis en place une évaluation pour mesurer la validité (ou grammaticalité en un sens) des phrases engendrées par rapport aux données d'entraînement. La procédure suivie pour créer une phrase dans le côté cible du corpus peut être interprétée comme un *transducteur fini pondéré* : chaque lettre d'entrée est conservée ou supprimée/doublée (selon qu'une compression ou une décompression est réalisée) en suivant des probabilités (Équation (5)) conditionnées sur le contexte des $n-1$ caractères précédents, contexte qui peut être associé à un état.

Notons T_r^{comp} les transducteurs pour la compression ($r \in [0,5;1]$), et T_r^{decomp} les transducteurs pour la décompression ($r \in [1;1,5]$). Grâce à eux, nous sommes capables de vérifier si une phrase produite par un modèle encodeur-décodeur est « correcte » par rapport à la procédure suivie pour créer le côté cible de notre corpus artificiel pour la compression de phrase. Une phrase $\hat{y} = \text{LenInit}_{r=0,6}(x)$, engendrée par LenInit à partir d'une phrase source x avec un taux de compression visé $r = 0,6$, est correcte si elle appartient à $\text{Im}(T_{r=0,6}^{comp} \circ x)$, l'ensemble des phrases cibles qui peuvent être obtenue depuis x par l'automate de compression correspondant (comme à chaque étape du décodage LenInit peut a priori engendrer n'importe quel caractère de l'alphabet, il est tout à fait possible que $\hat{y}$ ne soit pas dans le langage rationnel image du transducteur).

Nous pouvons aussi calculer la probabilité associée à une paire (x, y) formée d'une phrase source et d'une phrase cible correcte : $T_r^{comp}(x, y)$ (resp. $T_r^{decomp}(x, y)$), qui correspond à la probabilité

4. Les cas où $|l_{\hat{y}} - \lfloor r \times l_x \rfloor| = 0$.

x	`la guerre c est l enfer .`
$\hat{y} = \text{LenInit}_{r=0,6}(x)$	`la gerre c est lenfre`
$\hat{y}'$	`la gerre c est lenfe`

TABLE 1: Exemple de phrase source x, phrase prédite (produite par LenInit) $\hat{y}$, et plus proche phrase parmi celles que le transducteur de compression pourrait engendrer à partir de la source $\hat{y}'$.

cumulée de tous les chemins dans T_r^{comp} (resp. T_r^{decomp}) qui transduisent x en y. Dans notre évaluation, nous avons utilisé la probabilité logarithmique négative divisée par la longueur de la phrase source, définissant les scores :

$$S_r^{comp}(x,y) = \frac{-\log T_r^{comp}(x,y)}{l_x}, \quad S_r^{decomp}(x,y) = \frac{-\log T_r^{decomp}(x,y)}{l_x}. \tag{8}$$

Pour les cas dans lesquels une phrase prédite $\hat{y} = \text{LenInit}_r(x)$ n'est pas correcte (au sens donné ci-dessus), nous recherchons la phrase $\hat{y}'$ la plus proche (en terme de distance d'édition) dans $\text{Im}(T_r^{comp} \circ x)$ (resp. $\text{Im}(T_r^{decomp} \circ x)$), et calculons $S_r^{comp}(x, \hat{y}')$ (resp. $S_r^{decomp}(x, \hat{y}')$). Le tableau 1 donne un exemple d'un tel triplet $(x, \hat{y}, \hat{y}')$.

3.2 Prédiction de longueur à partir des états cachés

Notre deuxième groupe d'expériences s'attache à comprendre comment la longueur est représentée dans les états cachés du décodeur, et comment la contrainte impliquée par LenInit agit pendant le décodage. Shi *et al.* (2016); Adi *et al.* (2016) ont déjà montré qu'une information de longueur se trouve dans ce genre de représentations dans les cas de l'auto-encodage et de la traduction. Néanmoins nos essais se placent dans un cadre de compression de phrase avec un taux visé variable ; dans ce cas, la longueur cible ne peut être déterminée par une relation constante à partir de la longueur source, et le décodeur doit intégrer des données extérieures.

En utilisant le modèle LenInit (Section 2.1) entraîné sur notre corpus de compression (Section 2.2), nous avons conçu une tâche de classification qui prédit – étant donné un état caché échantillonné à une certaine étape j du décodage – la longueur de la séquence restant à produire. Les classes de sortie correspondent à des valeurs de longueur, entre 0 et 149 (ce qui est supérieur à la plus grande longueur enregistrée dans le corpus).

Nous avons créé un ensemble de données pour cette tâche, en échantillonnant aléatoirement une fraction (1%) de tous les états cachés produits pendant le décodage (effectué pour divers objectifs de taux de compression) de phrases issues de l'ensemble d'entraînement du corpus de compression [5]. À chacun de ces états cachés a été associé le nombre de caractères de sortie engendrés après qu'il a été échantillonné (c'est-à-dire entre son étape j et la fin du décodage de sa phrase). Ainsi, nous avons obtenu des classes représentées selon leurs proportions naturelles.

5. LenInit a été utilisé pour transduire les mêmes phrases sur lesquelles il avait été entraîné.

3.3 Évolution de la probabilité de génération du caractère de fin de phrase

Notre troisième groupe d'expériences suit l'évolution des probabilités respectivement associées au symbole *fin-de-phrase* et à certaines marques de ponctuation (à savoir « . », « ! » et « ? ») au cours du décodage. Des essais semblables ont été menés par Shi *et al.* (2016), sans trouver de progression régulière. Nous souhaitons vérifier si le cadre de la compression de phrase amène un changement à ce niveau.

Nous utilisons la même configuration que précédemment pour le modèle encodeur-décodeur (LenInit), que nous exécutons sur la partie développement du corpus de compression (Section 2.2). Les probabilités sont extraites des distributions sur le vocabulaire [6] créées à chaque étape du décodage.

4 Implémentation

Modèle de langue n-gramme au niveau caractère

Pour pouvoir produire le corpus expérimental de compression (Section 2.2), nous avons implémenté un modèle de langue 4-gramme au niveau caractère sous la forme d'un *perceptron multicouche* (Bengio *et al.*, 2003) avec une seule couche cachée (de dimension 128). Il a été entraîné pendant 4 époques sur 60 418 phrases en français de Tatoeba, en utilisant *Adam* (Kingma & Ba, 2015) avec son paramétrage standard : $\alpha = 0,0005$, $\beta_1 = 0,9$, $\beta_2 = 0,999$, $eps = 10^{-8}$.

LenInit et variantes

Nous avons implémenté le modèle LenInit (Section 2.1) comme un bi-*GRU* (Cho *et al.*, 2014) (dimension de plongement = 20, dimension cachée = 300, dimension de L_{param} = 300), entraîné pendant 1 époque sur 60 418 phrases en français de Tatoeba, en utilisant Adam (paramétrage standard).

Les variantes LenInit2 et LenInit3 ont les mêmes caractéristiques (en particulier, le nombre de paramètres dans L_{param} est le même).

Évaluation des phrases engendrées

Les transducteurs finis pondérés utilisés pour évaluer la validité des phrases engendrées ont été implémentés à l'aide de la librairie *Pynini* (Gorman, 2016).

Classification de la longueur future

Le classificateur pour la longueur future est un perceptron multicouche avec une seule couche cachée (de dimension 300, égale à la dimension d'entrée) activée par la fonction *ReLU*. Il a été entraîné pendant 5 époques sur 17 112 vecteurs échantillonnés, en utilisant Adam (paramétrage standard).

6. Pour notre modèle, le vocabulaire est l'ensemble des caractères utilisés

Les vecteurs des états cachés de l'ensemble de données pour la classification ont été produits en décodant des phrases de l'ensemble d'entraînement du corpus de compression : pour ces transductions nous avons utilisé LenInit, et pour chacune d'elles nous avons pris un taux de compression objectif uniformément échantillonné dans $[0,5; 1,5]$. Cette répartition aléatoire a été choisie afin de limiter l'introduction de biais dans la tâche.

5 Résultats

5.1 Compression/décompression avec LenInit

Avant de réaliser nos expériences autour des mécanismes de détermination de la longueur, nous avons testé LenInit sur le corpus artificiel de compression de phrase. Le tableau 2 présente quelques exemples de phrases issues de l'ensemble de développement, transduites avec différents taux de compression visés. Afin de vérifier la capacité du modèle à contrôler la longueur, nous avons transduit l'ensemble de développement selon plusieurs modalités :

1. en échantillonnant le taux visé r uniformément dans $[0,5; 1,5]$, ce qui permet de vérifier que la distribution des taux de compression obtenus (Figure 3a) est conforme à celle présente dans l'ensemble d'entraînement (Figure 1) (avec néanmoins une densité plus importante autour de 1, qui semble témoigner d'une tendance à reproduire la phrase source) ;
2. en fixant r à des valeurs présentes dans les données d'apprentissage ($0,6$, $1,0$ et $1,4$), ce qui permet d'observer des distributions de taux de compression effectivement centrées autour d'une valeur (Figure 3b) (notons toutefois que dans le cas $r = 1,4$, la gaussienne obtenue est davantage décalée par rapport au taux visé, probablement à cause de la difficulté pour l'encodeur-décodeur à manipuler les longues séquences) ;
3. en fixant r à des valeurs hors domaine (en dessous – $0,0$, $0,2$, $0,4$, et au dessus – $1,6$, $1,8$, $2,0$), ce qui montre (dans ce contexte au moins) l'incapacité de LenInit à généraliser son action pour des taux de compressions non-vus dans les données d'entraînement (Figures 3c, 3d).

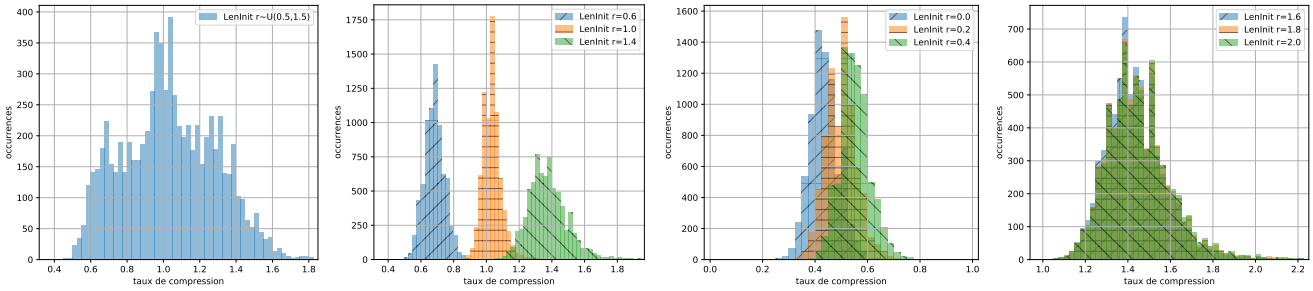
Ce dernier point n'est pas intuitif, dans la mesure où LenInit encode la valeur absolue de la longueur visée, et non le taux de compression (Équation (3)). Une valeur de r en dehors de $[0,5; 1,5]$ peut, selon la longueur de la phrase source, correspondre à une longueur cible (objectif) rencontrée lors de l'apprentissage. Cela laisse penser que le modèle de langue contenu dans le décodeur pourrait s'opposer et prévaloir face aux mécanismes de contrôle de longueur.

Le tableau 3 donne des mesures portant d'une part sur la précision du contrôle de longueur, et d'autre part sur la validité des phrases engendrées. Pour ce qui concerne la précision, LenInit2 et LenInit3 opèrent mieux que LenInit : les valeurs EAM et REQM montrent que les écarts aux valeurs de longueur visées sont moins grands dans l'ensemble, et les proportions d'instances proches des objectifs (EA=0, EA<5%, EA<10%) sont plus importantes. Il apparaît donc qu'un encodage par plongement permet plus de précision dans le contrôle de longueur qu'un encodage par la norme (qui en outre, comme précisé ci-dessus, ne garantit pas la fonctionnalité pour des valeurs hors domaine). Toutefois, l'encodage mixte de LenInit2 se montre plus efficace que le plongement pur de LenInit3, ce qui suggère que l'encodage continu apporte tout de même un bénéfice pour les valeurs peu fréquentes.

Nous observons également que LenInit réussit mieux à conserver la longueur ($r = 1$) qu'à compresser ou décompresser. Il est intéressant de noter que seulement 8,6% des phrases sont correctement recopiées par $\text{LenInit}_{r=1,0}$, quoique 25,1% des sorties soient de longueur exactes.

Source	`comment tom va t il ?`
LenInit$_{r=0,6}$	`commnt m a il ?`
LenInit$_{r=1,0}$	`comment tom va ti l ?`
LenInit$_{r=1,4}$	`commment tomm va ti l ??`
Source	`tout le monde a bord !`
LenInit$_{r=0,6}$	`tout lmode bor !`
LenInit$_{r=1,0}$	`tout le moodne a boord !!`
LenInit$_{r=1,4}$	`tout le moondde aa boord !!`
Source	`la guerre c est l enfer .`
LenInit$_{r=0,6}$	`la gerre c est lenfre`
LenInit$_{r=1,0}$	`la gurrre r c est l enfre .`
LenInit$_{r=1,4}$	`la guurerre c eesst l enfefre ..`

TABLE 2: Exemples de phrases transduites par LenInit (taux $0,6$, $1,0$ et $1,4$).



(a) $r \sim \mathcal{U}(0,5;1,5)$ (b) $r = 0,6;1,0;1,4$ (c) $r = 0,0;0,2;0,4$ (d) $r = 1,6;1,8;2,0$

FIGURE 3: Histogrammes des taux de compression dans l'ensemble de développement, transduit par LenInit selon différentes modalités. Les distributions se recouvrent pour les essais hors domaine.

Modèle	EAM	REQM	EA=0	EA<5%	EA<10%	corr.	S corr.	S inco.	S glob.
				$r \sim \mathcal{U}(0,5;1,5)$					
LenInit	8,4%	0,108	9,6%	35%	67%	16,4%	0,35	0,44	0,42
LenInit2	6,1%	0,086	18,5%	53%	82%	22,2%	0,39	0,47	0,45
LenInit3	7,7%	0,119	14,7%	45%	74%	19,8%	0,36	0,45	0,43
Oracle (T)	6,1%	0,080	17,0%	51%	80%	100%	0,37	-	0,37
				r fixé					
LenInit$_{r=0,6}$	8,6%	0,102	6,0%	29%	64%	15,5%	0,52	0,56	0,56
LenInit$_{r=1,0}$	4,7%	0,071	25,1%	62%	89%	8,6%	0	-	-
LenInit$_{r=1,4}$	9,9%	0,131	10,3%	31%	59%	6,1%	0,54	0,55	0,55
RNN$_{r=0,6}$	10,9%	0,195	7,30%	29%	58%	16,4%	0,38	0,47	0,45

TABLE 3: Mesures sur la précision du contrôle de longueur et sur la validité des phrases produites (Section 3.1). corr., S corr., S inco., S glob. indiquent respectivement la proportion de phrases correctes, et le score moyen attribué aux phrases correctes, incorrectes, et à l'ensemble des phrases. L'oracle est le transducteur fini pondéré qui a engendré le corpus.

Les scores de validité des phrases sont comparables, entre les variantes de LenInit et l'oracle (le transducteur utilisé pour générer le corpus).

À titre de comparaison nous avons aussi entraîné un encodeur-décodeur RNN classique sur un corpus comparable mais ne contenant que des phrases compressées pour $r = 0,6$. Les scores de validité indiquent que le RNN a appris un modèle de langue adapté, et nous notons une précision un peu moins importante que celle de $\mathrm{LenInit}_{r=0,6}$.

5.2 Prédiction de la longueur future

Nous avons testé notre classificateur sur l'ensemble de développement du corpus de prédiction de la longueur future (Section 3.2), qui contient $n = 2169$ états cachés échantillonnés. La figure 4a montre la distribution de la différence entre la longueur prédite par le modèle et la longueur de référence (récupérée expérimentalement lors de la création de l'ensemble de données). Cette distribution est approximativement centrée autour de 0 et nous avons calculé les valeurs EAM et REQM comme suit :

$$\text{EAM} = \frac{1}{n}\sum_{i=1}^{n}|\hat{l}_i - l_i| = 3,08, \quad \text{REQM} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{l}_i - l_i)^2} = 4,94, \tag{9}$$

où $\hat{l}_i$ et l_i sont respectivement la longueur prédite et la longueur de référence pour le i-ème échantillon.

Notons que ce résultat confirme l'existence dans un état caché du décodeur de données qui représentent spécifiquement la longueur à venir, et qui ne peuvent dériver directement d'un reliquat d'information relatif à la phrase d'entrée. En effet, puisque le modèle LenInit qui a engendré les états cachés avait reçu des objectifs choisis aléatoirement (uniformément dans $[0,5; 1,5]$), le classificateur ne devrait pas avoir pu établir de corrélation systématique entre la longueur d'entrée et la longueur de sortie.

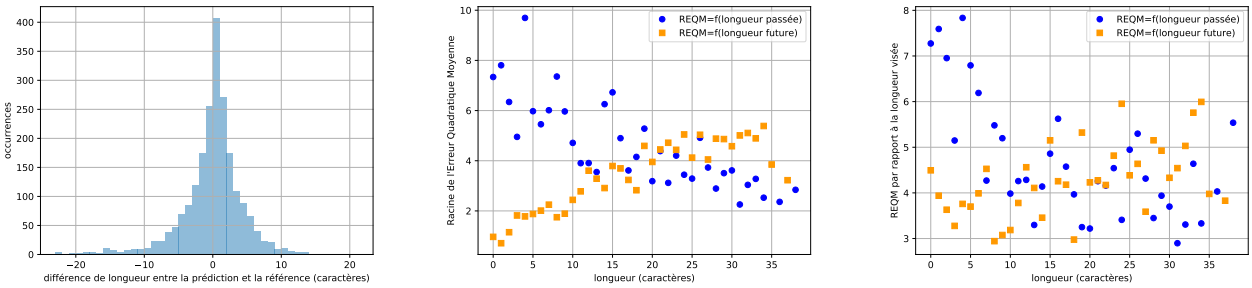
La figure 4b montre l'évolution de la mesure REQM en fonction de la position de l'état caché dans la séquence de sortie. L'indicateur d'erreur est calculé selon :

$$\text{REQM}(k) = \sqrt{\frac{1}{|I_k|}\sum_{i\in I_k}(\hat{l}_i - l_i)^2}, \tag{10}$$

où I_k est l'ensemble d'indices qui décrit les instances de l'ensemble de développement qui ont été échantillonnées à la position k dans leur phrase de sortie ; les deux séries sur la figure se distinguent en mesurant k soit depuis le début de la phrase, soit avant sa fin. La tendance suggère que la précision de la prédiction augmente au fil du décodage.

Similairement, la figure 4c montre l'évolution au cours du décodage d'une mesure de type REQM, cette fois calculée par rapport à la longueur visée : dans l'équation (10), l_i ne correspond plus à la longueur future de référence (celle obtenue en pratique), mais à la longueur future visée (celle qu'il faudrait produire pour atteindre l'objectif de compression donné par r). Les profils sont moins nets dans ce cas, mais nous constatons que l'erreur en fin de phrase est plus importante (ce qui découle logiquement de l'imprécision du contrôle exercé par LenInit), et que l'erreur en début de phrase est étonnamment haute, étant donné que la longueur visée est intégrée dans le premier état caché.

Ces observations peuvent être interprétées comme des signes que l'objectif de longueur plongé dans l'état initial coexiste durant le décodage avec un autre mécanisme de contrôle de la longueur, propre au décodeur, qui est influencé par les caractères progressivement engendrés.

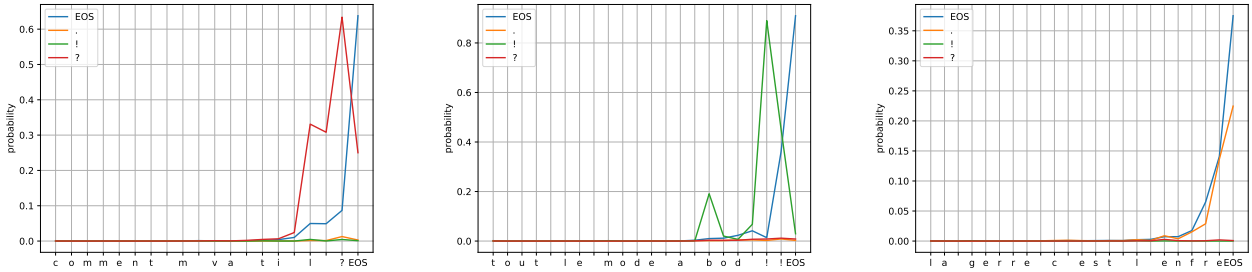


(a) Différence par rapport à la longueur de référence

(b) REQM par rapport à la longueur de référence

(c) REQM par rapport à la longueur visée

FIGURE 4: (a) Histogramme de la différence $(\hat{l}_i - l_i)$, sur l'ensemble de développement. (b), (c) Évolution au cours du décodage de l'erreur de la longueur prédite par rapport à la longueur de référence et par rapport à la longueur visée. Les marqueurs carrés correspondent à l'erreur en fonction du nombre d'étapes avant la génération de *fin-de-phrase*. Les marqueurs ronds correspondent à l'erreur en fonction du nombre d'étapes après *début-de-phrase*.



(a) Source :
`comment tom va t il ?`

(b) Source :
`tout le monde a bord !`

(c) Source :
`la guerre c est l enfer .`

FIGURE 5: Évolution au cours du décodage de la probabilité attribuée par $\text{LenInit}_{r=0,8}$ aux caractères « . », « ! », « ? » et *fin-de-phrase*, pour trois phrases exemples.

5.3 Évolution de la probabilité des caractères de fin de phrase

Après avoir analysé l'information de longueur présente dans les états cachés, nous nous sommes penchés sur ce que le modèle de langue du décodeur en faisait. La figure 5 donne l'évolution de la probabilité attribuée par le modèle LenInit (appliqué avec un taux de compression visé $r = 0,8$) aux caractères « . », « ! », « ? » et *fin-de-phrase*, pendant le décodage de phrases provenant de Tatoeba (présentes dans l'ensemble de développement du corpus de compression). Après examen d'un large éventail de tels graphes, il apparaît que la hausse qui provoque la génération de *fin-de-phrase* est généralement très abrupte, et qu'elle succède la plupart du temps à la génération d'une marque de ponctuation, qui elle même n'a pas eu lieu après une montée régulière ou étagée de probabilité. Ainsi, contrairement aux mesures de la Section 5.2, la probabilité des caractères de fin de phrase semble être un signal final à travers lequel il est difficile de suivre la progression du processus de contrôle de la longueur.

6 Travaux connexes

La simplification de phrase et la compression de phrase sont des domaines bien étudiés du traitement automatique des langues, qui ont bénéficié des avancées majeures apportées par la traduction automatique aux méthodes de transduction. Aux approches pionnières fondées sur les règles (Cohn & Lapata, 2008) ont succédé - dans la suite de *Moses* (Koehn *et al.*, 2007) - des modèles statistiques comme ceux de Specia (2010); Wubben *et al.* (2012). Puis, grâce au cadre des réseaux neuronaux récurrents (Sutskever *et al.*, 2014; Bahdanau *et al.*, 2015), Rush *et al.* (2015) ont implémenté pour la tâche un système encodeur-décodeur avec attention. Enfin plus récemment, des travaux tels que ceux de Zhao *et al.* (2018); Takase & Okazaki (2019) ont adapté l'architecture *Transformer*.

Quoique souvent comparée à la traduction, la simplification de phrase peut également être décrite du point de vue du transfert de style, dans la mesure où elle vise à reformuler la même signification avec un tour moins complexe (et plus concis parfois). La littérature sur le transfert de style propose généralement de travailler avec des représentations dans un espace continu, et d'y séparer le contenu sémantique des dimensions stylistiques. Ainsi Hu *et al.* (2017) utilisent un *auto-encodeur variationnel* avec des représentations *démêlées* pour engendrer des phrases dont les attributs (parmi lesquels la longueur) sont contrôlés. La dissociation entre la représentation style et celle du fond sémantique est réalisée dans ce cas par apprentissage adverse. En comparaison, le modèle LenInit de Kikuchi *et al.* (2016) utilise un paramètre pour encoder la longueur voulue et le concatène à la représentation de la phrase initiale, sans employer de mécanisme pour assurer que celle-ci ne contienne pas d'information de longueur. Cela rejoint sur le principe certaines méthodes qui contrôlent la catégorie de la phrase produite en ajoutant un symbole spécifique à la fin de la phrase d'entrée (pour la formalité (Sennrich *et al.*, 2016), ou le domaine pour (Kobus *et al.*, 2017)).

Nous suivons dans cet article les études de sondage des représentations, qui ont détecté certaines informations de surfaces (longueur entre autres) dans les plongements de phrases (Adi *et al.*, 2016; Conneau *et al.*, 2018). Nos expériences sur la prédiction de longueur future et le suivi de probabilité au cours du décodage sont en partie similaires à celles de Shi *et al.* (2016) (dans le cadre de la compression de phrase toutefois).

7 Conclusion

Nous avons réimplémenté la méthode LenInit, et avons étudié son efficacité et ses mécanismes de régulation de longueur pour une tâche de compression/décompression sur un corpus artificiel. Il semble que deux influences s'opposent au cours du décodage : d'une part l'objectif explicite de longueur, et d'autre part la contrainte de produire une phrase grammaticalement correcte exercée par le modèle de langue du décodeur. Nous avons pu déceler ces influences dans l'information contenue dans les états cachés, mais sans encore comprendre exactement comment elles agissent. Si LenInit exerce bien un contrôle sur la longueur, celui-ci n'est pas étroit mais plutôt probabiliste (d'après des expériences où pourtant la consigne pourrait toujours être précisément respectée), et peut être influencé par le choix d'encodage. À l'avenir, il pourrait être intéressant de tester si la contrôlabilité dépend de certaines caractéristiques de la phrase d'entrée, ou si la distribution des types d'opérations choisis par le modèle change pendant le décodage.

Références

ADI Y., KERMANY E., BELINKOV Y., LAVI O. & GOLDBERG Y. (2016). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, **abs/1608.04207**.

AZIZ W., DE SOUSA S. C. M. & SPECIA L. (2012). Cross-lingual sentence compression for subtitles. In *16th Annual Conference of the European Association for Machine Translation*, EAMT, p. 103–110, Trento, Italy.

BAHDANAU D., CHO K. & BENGIO Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. BENGIO & Y. LECUN, Éds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

BENGIO Y., DUCHARME R., VINCENT P. & JANVIN C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, **3**, 1137–1155.

CHO K., VAN MERRIËNBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1179.

COHN T. & LAPATA M. (2008). Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics*, (COLING 2008), p. 137–144, Manchester, UK : Coling 2008 Organizing Committee.

CONNEAU A., KRUSZEWSKI G., LAMPLE G., BARRAULT L. & BARONI M. (2018). What you can cram into a single $& !#* vector : Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2126–2136 : Association for Computational Linguistics.

GORMAN K. (2016). Pynini : A Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, p. 75–80, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/W16-2409.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780. DOI : 10.1162/neco.1997.9.8.1735.

HORN C., MANDUCA C. & KAUCHAK D. (2014). Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 458–463, Baltimore, Maryland : Association for Computational Linguistics. DOI : 10.3115/v1/P14-2075.

HU Z., YANG Z., LIANG X., SALAKHUTDINOV R. & XING E. P. (2017). Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 1587–1596 : JMLR.org.

KIKUCHI Y., NEUBIG G., SASANO R., TAKAMURA H. & OKUMURA M. (2016). Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1328–1338 : Association for Computational Linguistics. DOI : 10.18653/v1/D16-1140.

KINGMA D. P. & BA J. (2015). Adam : A method for stochastic optimization. In Y. BENGIO & Y. LECUN, Éds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

KNIGHT K. & MARCU D. (2002). Summarization beyond sentence extraction : A probabilistic approach to sentence compression. *Artif. Intell.*, **139**(1), 91–107. DOI : 10.1016/S0004-3702(02)00222-9.

KOBUS C., CREGO J. & SENELLART J. (2017). Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, p. 372–378, Varna, Bulgaria : INCOMA Ltd. DOI : 10.26615/978-954-452-049-6_049.

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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, p. 177–180, Prague, Czech Republic : Association for Computational Linguistics.

MALLINSON J., SENNRICH R. & LAPATA M. (2018). Sentence compression for arbitrary languages via multilingual pivoting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2453–2464 : Association for Computational Linguistics.

RUSH A. M., CHOPRA S. & WESTON J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 379–389 : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1044.

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

SHI X., KNIGHT K. & YURET D. (2016). Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 2278–2282, Austin, Texas : Association for Computational Linguistics. DOI : 10.18653/v1/D16-1248.

SPECIA L. (2010). Translating from complex to simplified sentences. *Lecture Notes in Computer Science*, **6001**, 30–39. DOI : 10.1007/978-3-642-12320-7_5.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. In Z. GHAHRAMANI, M. WELLING, C. CORTES, N. D. LAWRENCE & K. Q. WEINBERGER, Éds., *Advances in Neural Information Processing Systems 27*, p. 3104–3112. Curran Associates, Inc.

TAKASE S. & OKAZAKI N. (2019). Positional encoding to control output sequence length. *CoRR*, **abs/1904.07418**.

WUBBEN S., VAN DEN BOSCH A. & KRAHMER E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1015–1024 : Association for Computational Linguistics.

ZHAO S., MENG R., HE D., SAPTONO A. & PARMANTO B. (2018). Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 3164–3173 : Association for Computational Linguistics.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Exploitation de modèles distributionnels pour l'étude de la nomination dans un corpus d'interviews politiques

Manon Cassier[1, 2]
(1) CY Cergy-Paris Université, AGORA, 33 Boulevard du port, 95011 Cergy, France
(2) Institut National des Langues et Civilisations Orientales, ERTIM, 2 Rue de Lille, 75007 Paris, France

manon.cassier@u-cergy.fr

RÉSUMÉ

En analyse de discours (AD), la nomination désigne la recatégorisation du référent par le locuteur à travers l'usage d'un nouveau nom ou d'un nom modifié. Parfois utilisé pour influencer l'autre sur sa vision de voir le monde, ce phénomène sert d'indice sur l'idéologie du locuteur voire, en contexte adéquat, sur son affiliation politique. L'AD ne dispose pas à ce jour d'outils en mesure d'appréhender efficacement ce qui relève ou non de l'idéologie ou d'une visée argumentative face à une simple réutilisation de mots dont le sens est déjà consensuel. Dans le cadre d'une thèse entre AD et TAL, nous nous intéressons à l'exploitation de modèles distributionnels pour repérer de manière automatique ces variations de sens en discours dans un corpus d'interviews politiques. Dans cet article, nous nous interrogeons sur l'impact de leurs paramètres d'entraînement pour de la désambiguïsation lexicale et explorons une méthode de représentation de la variation sémantique interdiscursive.

ABSTRACT

Speaker-specific semantic variation representations using vector space models.

The French Discourse Analysis (DA) concept of "Nomination" designates the cases of the speaker updating an entity's identity by using words in a particular way that involves its opinion. It sometimes constitutes a good way to gain access to the person's ideology and to predict its political stripe. Several tools are already used by discourse analysts, but they do not provide to properly represent the nomination semantic characteristics. In our work, we question word embeddings based models benefits to automatically detect these speaker-specific semantic changes. In this paper, we investigate the role of training features for a semantic oriented task and test some approaches to represent semantic variation.

MOTS-CLÉS : Nomination, analyse du discours (AD), désambiguïsation lexicale, modèles distributionnels.

KEYWORDS: Nomination, Political Discourse Analysis, Word Sense Induction (WSI), Word Sense Disambiguation (WSD), Word Embeddings Based Models.

1 Introduction

Le concept de nomination est étudié en analyse du discours (AD) comme la réassignation sémantique et référentielle en discours d'un mot, dans certains cas à des fins de valorisation idéologique et de persuasion de l'auditoire. Parfois considéré comme une pratique de l'argumentation, l'acte de nomination est réalisé dans des discussions autour de sujets propices à la controverse, dans des

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 71–84.
HAL : hal-02786188.

contextes socio-politiques qui portent au conflit, au sein desquels il sert à qualifier l'objet du désaccord (Koren, 2016). Le locuteur manipule alors le lexique de manière à négocier le sens du mot et imposer sa représentation du référent en influençant l'autre. Gauthier (2016) montre par exemple comment l'utilisation par le gouvernement canadien du mot « boycott plutôt que « grève » pour désigner la cessation d'activité des associations étudiantes contre la hausse des frais de scolarité sert une stratégie d'évitement du débat par une minimisation de l'impact de l'événement.

L'analyse de discours dispose déjà, à ce jour, de différents outils pour l'exploration de textes et la mesure des spécificités lexicales (i.e. le sur- ou sous-emploi d'une forme par un locuteur ou par un genre de discours particulier) (Tournier, 1981). Ces outils reposent tous sur des méthodes de calculs statistiques qui permettent par exemple d'accéder à la fréquence des cooccurents ou des segments répétés, mais qui ne permettent pas d'appréhender l'aspect sémantique indispensable pour l'étude de la nomination. En effet, s'il est possible d'observer des tendances à l'usage de certains mots choisis plutôt que d'autres (Marchand & Ratinaud (2012) montrent par exemple un sur-emploi des formes « écologique » et « équitable » dans le discours de Ségolène Royal lors des primaires socialistes de 2011), seule une analyse humaine permet à ce jour de discriminer les usages dits « neutres » de ceux qui relèveraient de la nomination (i.e. dont le sens serait spécifique à l'idéologie du locuteur).

Le projet ANR TALAD [1] se donne pour objectif de fournir à l'AD de nouveaux outils, adaptés des tâches du TAL, qui permettront d'assister plus efficacement la recherche de phénomènes tels que la nomination sur de gros volumes de textes.

Il s'agirait par exemple d'aider à l'identification, dans les exemples suivants [2], du candidat nomination « Europe ». Celui-ci est utilisé dans le premier cas pour référer à l'entité politique perçue comme bénéfique, mais réfère au contraire à une simple entité géographique dans le second pour contraster avec l'« Union européenne » qui endosse à sa place la référence à l'entité politique perçue comme négative.

Exemple 1. « Si la France veut continuer de se projeter dans le monde, elle doit avancer en **Europe**, rebâtir le projet européen avec détermination et là aussi ne rien céder à celles et ceux qui doutent. » (Discours d'Emmanuel Macron à Montpellier le 18/10/2016)

Exemple 2. « Donc l'Union européenne est une mauvaise chose, pas l'**Europe**. L'**Europe** c'est une entité, une réalité de civilisations, c'est une réalité géographique, historique. » (Interview de Marine Le Pen sur Europe1 le 01/05/2017)

Dans le cadre de notre thèse, nous nous focalisons en particulier sur l'exploration de méthodes distributionnelles pour mettre au jour les variations sémantiques inter-locuteurs au niveau du mot. En effet, de tels modèles ont déjà fait leurs preuves dans des travaux portés sur la sémantique des mots, notamment pour des tâches de désambiguïsation lexicale.

Néanmoins, notre problématique apporte un aspect qui, à notre connaissance, n'est pas encore abordé par la littérature existante, à savoir la mise au jour de sens nouveaux ou non consensuels (Pengam & Jackiewicz, 2019), et de variantes parfois si subtiles et peu fréquentes que même l'identification humaine en est rendue difficile. De même, notre objectif ne se résume pas à de la désambiguïsation lexicale, puisqu'il s'agit aussi de remarquer un déplacement du sens, parfois simplement d'un tour de parole à un autre (et donc au sein d'un même corpus). Enfin, notre intérêt final serait d'aboutir à une méthode qui puisse nous permettre de mesurer, sur le modèle des calculs de spécificités lexicales

1. https://anr.fr/Project-ANR-17-CE38-0012
2. Exemples tirés du corpus *Reticular* présenté en section 3.1

(Tournier, 1981), des spécificités sémantiques capables de caractériser chaque acteur en fonction du sens qu'il accorde à ses mots.

Nous commençons cet article par un état des lieux des outils déjà exploitables ou exploités en AD et présentons leurs lacunes pour des études focalisées sur la nomination en première partie de section 2. Nous présentons en seconde partie de cette même section un état de l'art des méthodes exploitées notamment pour la détection de la polysémie et la désambiguïsation lexicale desquelles nous nous inspirons pour notre travail. La section 3 détaille le dispositif expérimental employé dans cette étude, notamment les corpus et les méthodes de sélection des paramètres utilisés pour l'entraînement des modèles. La section 4 introduit une expérimentation centrée sur la prédiction d'un mot-pivot. Nous présentons ensuite la démarche que nous envisageons pour représenter la variation interdiscursive en section 5. La section 6 conclut et discute les choix méthodologiques et les poursuites envisagées.

2 État de l'art

2.1 Outiller l'analyse de discours

Les analystes du discours travaillent généralement autour de mots-pivots, sélectionnés car potentiellement vecteurs d'idéologie (Mazière, 2018). Ce type de méthode nécessite au préalable d'identifier le mot autour duquel il faudra construire un corpus qui réunira les contextes représentatifs du phénomène étudié. L'utilisation de logiciels de lexicométrie ou textométrie (e.g. Lexico, Iramuteq, Hyperbase, Alceste ou TXM pour ne citer que les plus utilisés en AD), en l'occurrence par le biais des concordanciers qu'ils proposent, peut faciliter la constitution de corpus, qui représente déjà en soi une étape assez chronophage du travail de l'analyste du discours. Néanmoins, l'utilisation de ces concordanciers seuls ne suffit pas pour repérer de nouvelles formes candidates susceptibles d'être concernées par la nomination lorsque le contexte nécessaire à leur reconnaissance est plus large que la fenêtre d'observation offerte par l'outil, ni lorsqu'il s'agit d'étudier des variantes lexicales pour une même nomination. Pengam & Jackiewicz (2019) montrent par exemple que la nomination autour de « musulman modéré » est susceptible d'entraîner des variantes telles que « islam modéré » ou « islamisme modéré » qu'il est nécessaire d'anticiper si elles n'apparaissent pas dans les mêmes contextes que la forme pivot pré-sélectionnée.

Aussi, le fait de travailler justement autour de formes pivots, généralement pré-sélectionnées pour leur caractère néologique (comme c'est le cas pour la collocation « musulman modéré ») et/ou polémique (comme c'est le cas pour « migrant » (Calabrese, 2018)) limite fortement l'étude de l'ensemble des nominations possibles et peut empêcher de détecter les cas les plus flagrants.

Ces logiciels, développés à l'origine pour des études en statistiques textuelles, offrent chacun des fonctionnalités qui s'avèrent utiles pour les études menées en analyse de discours (Longhi, 2017). Les calculs de cooccurrences et la recherche de segments répétés (Lebart & Salem, 1994) peuvent par exemple assister la détection de nouvelles collocations comme pour « musulman modéré », qui donnent un indice sur le changement de sens du mot tête (ici « musulman »). La classification de Reinert (Reinert, 1993) offre la possibilité de distinguer les différentes thématiques abordées par un corpus, qui peuvent par exemple aider à déterminer les cadres de la nomination (i.e. les thèmes les plus probables de susciter l'emploi de nominations). Le calcul de spécificités lexicales (Tournier, 1981) permet de déterminer, pour un corpus découpé en sous-parties (distinguées par exemple selon le genre textuel, l'année ou le locuteur), les formes en sur- ou en sous-emploi dans ces sous-parties (i.e. les formes qui en sont les plus ou les moins réprésentatives). Ce type de calcul offre la possibilité, dans le cadre de l'analyse du discours politique, de déterminer rapidement le vocabulaire qui est le

plus spécifique à chaque politicien. En revanche, puisque ces calculs ne fonctionnent que sur la base de statistiques sur la fréquence des mots dans chaque corpus, sans prendre en compte le contexte d'apparition des mots, il est compliqué d'en tirer des conclusions au niveau sémantique. Un tri manuel des contextes une fois les spécificités lexicales identifiées est toujours indispensable.

Sans vouloir dispenser de cette étape nécessaire d'étude manuelle de contextes, ce travail se place dans l'objectif de proposer un outil plus complet pour assister l'analyse de discours (et plus spécifiquement l'analyse de la nomination) en intégrant l'aspect sémantique à l'extraction de contextes candidats. L'étude présentée dans ce papier consiste en l'exploration de méthodes distributionnelles pour le repérage automatique de candidats nominations, dont nous présentons un état de l'art dans la section suivante.

2.2 Les modèles distributionnels au service de la désambiguïsation lexicale

Les représentations vectorielles de mots se fondent sur l'hypothèse distributionnelle introduite par Harris (1954) et Firth (1957) selon laquelle le sens d'un mot peut être déterminé grâce aux contextes dans lesquels il apparaît, et que les mots qui apparaissent dans des contextes similaires sont sémantiquement proches. Les pratiques de l'AD abordées dans la section précédente montrent également l'importance de ces contextes pour les analystes du discours qui fondent leur travail sur l'étude des cooccurrences et des segments répétés. Les différents modèles prédictifs qui ont pu voir le jour au cours de la dernière décennie suscitent toujours un grand engouement pour leurs performances sur de nombreuses tâches du TAL et leur capacité à fournir une représentation sémantique des mots en les situant les uns par rapport aux autres dans un espace vectoriel dense.

Les outils comme Word2Vec (Mikolov *et al.*, 2013) jouissent désormais d'une documentation suffisamment complète pour être éprouvés dans des études purement linguistiques, les mécanismes en jeu dans l'apprentissage des représentations faisant régulièrement l'objet de travaux qui les ont rendus beaucoup plus compréhensibles qu'à leurs débuts (e.g. Levy & Goldberg (2014); Levy *et al.* (2015); Patel & Bhattacharyya (2017) sur l'influence des paramètres d'apprentissage ; Pierrejean & Tanguy (2018, 2019) sur la variabilité des représentations et l'instabilité des modèles). En outre, les modèles prédictifs qu'ils proposent suscitent beaucoup d'intérêt pour la résolution de tâches focalisées sur des aspects sémantiques telles que l'analogie (i.e. reconstitution de paires de concepts qui entretiennent le même type de relation que les concepts d'une paire exemple, e.g. de type *Pays-Capitale* : Paris est à la France ce que Madrid est à l'Espagne). Enfin, les travaux de Baroni *et al.* (2014) et Levy *et al.* (2015) ont montré que ces modèles prédictifs, entraînés avec des paramètres adaptés à la tâche et au corpus d'étude, peuvent obtenir de meilleurs résultats que les approches à base de comptes (e.g. *Positive Pointwise Mutual Information* (PPMI), *Latent Semantic Analysis* (LSA)).

Néanmoins, ces modèles présentent le défaut de ne fournir qu'une seule représentation vectorielle par forme ou par mot, en encodant tous les sens possibles de cette forme ou de ce mot en un vecteur unique lorsqu'il serait préférable d'avoir un vecteur par contexte. Différentes méthodes ont déjà été explorées pour transformer ces vecteurs uniques en vecteurs de sens, notamment dans le cadre de travaux en induction de sens (*Word Sense Induction* ou WSI) et en désambiguïsation lexicale (*Word Sense Disambiguation* ou WSD)(Ruas *et al.*, 2019; Pelevina *et al.*, 2017).

La plupart des systèmes de WSI et WSD se basent sur un inventaire de sens, qui liste les différents sens possibles de chaque mot, pour calculer la probabilité d'un contexte de relever d'un des sens répertoriés pour un mot. De nombreux travaux exploitent par exemple des bases de données comme *WordNet* (Vial *et al.*, 2017) ou *BabelNet* (Dongsuk *et al.*, 2018) qui associent des groupes de mots

qui entretiennent des relations sémantiques (e.g. hypéronymie, méronymie) au sein de *synsets*. Les ressources de ce type sont généralement moins développées pour le français. En outre, dans le cas de la nomination, le sens du mot n'est pas forcément attesté dans un dictionnaire, ni partagé par l'ensemble de la communauté linguistique puisqu'il dépend de l'appréciation du locuteur du référent qu'il nomme. La ressource la plus exhaustive possible resterait donc inexploitable pour distinguer des variantes de sens ou des représentations du référent complètement nouvelles.

Pour contourner ce problème, Pelevina *et al.* (2017) proposent une méthode d'acquisition automatique d'inventaire de sens à partir de textes non annotés via la création d'*ego-networks* (i.e. identification de clusters de sens autour d'un mot pivot). Ces réseaux sont ensuite exploités pour calculer un vecteur de sens comme vecteur moyen des vecteurs de tous les mots appartenant au cluster. Les vecteurs de sens induits sont ensuite comparés aux vecteurs de contexte des mots à désambiguïser.

Nous notons, pour la suite de notre travail, que la plupart de ces approches sont évaluées sur des *benchmarks* pour des tâches précises comme la similarité ou l'analogie. Ces *benchmarks*, souvent constitués pour l'anglais, évaluent des relations sémantiques tirées du domaine général qui diffèrent beaucoup de celles que nous pouvons observer dans des corpus de spécialité comme les interviews politiques. Les particularités de notre tâche (i.e. évaluée sur du français, dans un corpus de spécialité, avec l'objectif supplémentaire de comparer les représentations interdiscursives) devront nous questionner sur l'adaptation de méthodes d'évaluation, sur le modèle par exemple de Bloem *et al.* (2019) qui proposent une mesure de cohérence des représentations apprises dans des corpus de spécialité.

Au regard de cet état de l'art, nous présentons dans cet article notre réflexion sur l'exploitabilité de modèles distributionnels pour représenter et détecter la variation sémantique interdiscursive au niveau du mot (plus précisément le nom) et leur évaluation sur un corpus d'interviews politiques françaises.

3 Dispositif expérimental

À ce stade de la thèse, nous avons choisi de concentrer nos expérimentations sur des modèles entraînés avec Word2Vec (Mikolov *et al.*, 2013). Ce choix s'explique notamment par l'accès facilité à de la documentation claire, la possibilité d'entraîner soi-même ses modèles pour un coût relativement faible face à des modèles de réseaux de neurones très gourmands en GPU, mais aussi par la multiplication de travaux qui ont déjà prouvé un fonctionnement suffisant sur des corpus limités (Bloem *et al.*, 2019). Nous envisageons, si les premiers résultats sont satisfaisants et que le temps nous le permet, de poursuivre nos explorations avec les modèles à base de réseaux de neurones plus élaborés qui font aujourd'hui état de l'art pour de nombreuses tâches du TAL (e.g. BERT, ELMo).

Comme nous l'expliquons dans la section 1, notre problématique nécessite de repérer à la fois les usages « inhabituels » de noms (i.e. dont le sens diffère de celui qui serait consensuel à l'ensemble des locuteurs de la langue et qui serait disponible dans une ressource dictionnairique ou inventaire de sens), mais aussi les variantes de sens d'un acteur politique à un autre.

Suivant ces deux contraintes, nos expérimentations sont basées sur deux corpus de genres différents. Le corpus *Reticular*, que nous présentons dans la sous-section suivante rassemble des interviews politiques et nous sert à construire les représentations subjectives à analyser (qui contiennent potentiellement des nominations). Pour comparer ces représentations à un usage plus « neutre », nous utilisons un corpus constitué des articles de la version française de Wikipédia de 2008 [3].

3. Corpus WikipediaFR2008 extrait par l'équipe CLLE-ERSS de l'Université de Toulouse Jean Jaurès, disponible à l'adresse suivante : http://redac.univ-tlse2.fr/corpus/wikipedia.html

3.1 Le corpus *Reticular*

Pour notre étude, nous nous basons sur un corpus d'interviews transcrites à la main. Ce corpus couvre 3166 interviews de 561 personnalités publiques données pour des émissions de radio dans le contexte des élections présidentielles françaises de 2017. Le corpus couvre la période de juin 2016 à décembre 2017 et compte environ 11 millions de mots.

Chacune des interviews du corpus est accompagnée de métadonnées indiquant la date et l'heure de l'interview et le nom de l'interviewé. Une interview est découpée en tours de parole, avec une ligne par tour de parole, alternant les questions et réactions du journaliste et de l'interviewé. La répartition des interviews n'est pas homogène, les émissions de radio accueillant à la fois des politiciens et des personnes extérieures à la sphère politique susceptibles d'éclairer un débat (e.g. économistes, dirigeants d'entreprise, etc.). La majorité des interviews sont néanmoins partagées par des têtes de listes et leur directeur de campagne.

Bien qu'il s'agisse d'un corpus transcrit de l'oral, le corpus *Reticular* ne contient que le texte des interviews, sans marques d'hésitations. De même, la plupart des répétitions, habituelles dans les corpus oraux, n'ont pas été transcrites. Aussi, les marques de ponctuation n'ont été utilisées par les transcripteurs que pour découper le texte en phrases cohérentes. Nous devons donc préciser qu'il nous est impossible d'utiliser ces marques à des fins d'analyse sémantique.

Pour toutes les expérimentations présentées dans les sections suivantes, nous utilisons une version lemmatisée avec Treetagger (Schmid, 2013) du corpus (en conservant les formes en lieu et place des lemmes non reconnus) à laquelle nous retirons toutes les marques de ponctuations. Nous ne considérons donc pas les phrases, mais les tours de parole marqués dans le corpus par des sauts de ligne.

Les mêmes prétraitements sont appliqués au corpus *Wikipédia*, en plus du retrait des caractères spéciaux qui n'apparaissent pas dans le corpus des interviews. Aussi, pour limiter l'impact des balises, hyperliens et phrases typiques de l'encyclopédie en ligne (e.g. « Ceci est un article qui concerne... »), nous faisons un tri grossier en retirant les lignes de moins de 10 mots.

3.2 Méthode de sélection des paramètres d'entraînement

Dans un premier temps, comme pour répondre à toute tâche de TAL, notre travail nécessite de déterminer les paramètres d'entraînement de nos modèles qui auront l'impact le plus positif sur les représentations distributionnelles qui en découleront. Certains travaux ont investigué l'incidence du choix des paramètres d'entraînement sur la performance des modèles. Patel & Bhattacharyya (2017) montrent par exemple que les représentations sont plus fiables pour des tâches de similarité, d'analogie et de classification lorsque le nombre de dimensions dépasse un seuil qui dépend du nombre de noeuds du plus large cluster disponible dans un graphe de similarités construit à partir de la matrice de cooccurrences du corpus. En dessous de ce seuil, le modèle n'est pas capable d'encoder tous les liens entre les mots du corpus et donne donc de moins bons résultats. En revanche, la performance du modèle a tendance à se stabiliser pour un nombre de dimensions supérieur à ce seuil. Les travaux de Levy & Goldberg (2014) et Levy *et al.* (2015) démontrent de leur côté qu'une taille de fenêtre petite (i.e. pour laquelle on réduit le contexte à prendre en compte autour du mot lors de l'apprentissage du modèle) permet de mieux représenter les caractéristiques syntaxiques du mot (e.g. quelle catégorie grammaticale a le plus de chance de se retrouver directement en contact avec le mot cible) alors qu'un contexte plus grand est capable d'encoder les aspects plus sémantiques (e.g. quel synonyme a le plus

de chance de se retrouver à la place de ce mot) puisqu'il a accès aux cooccurents plus éloignés du mot.

Puisque l'étude de la nomination demande de s'intéresser essentiellement à la catégorie nominale, nous souhaitons en premier lieu éprouver cette remarque en vérifiant si la taille de fenêtre sélectionnée lors de l'entraînement de nos modèles impacte particulièrement la prédiction des noms et des entités nommées (Nouvel *et al.*, 2015). Pour ce faire, nous exploitons simplement la fonction principale de l'architecture CBOW (que la littérature présentée dans la section 2 reconnaît comme la plus efficace pour des tâches de désambiguïsation lexicale), à savoir la prédiction d'un mot à partir d'un contexte.

À partir du corpus lemmatisé de *Wikipédia*, dans lequel nous ajoutons à chaque mot l'étiquette de sa catégorie grammaticale récupérée avec Treetagger, nous entraînons deux modèles avec les paramètres par défaut de Word2Vec (i.e. 100 dimensions avec un *negative sampling* et 5 itérations sur le corpus) en ne faisant varier que la taille de fenêtre (i.e. la taille du contexte considéré), fixée à 5 pour le premier modèle et à 15 pour le second. Nous fixons la fréquence minimale des mots à prendre en compte dans le calcul à 10 pour limiter l'impact des mots rares, susceptibles d'être répétés plusieurs fois dans les articles de Wikipédia.

Nous évaluons ensuite la capacité des deux modèles à prédire chaque catégorie grammaticale en fonction du contexte qui leur est fourni dans un système de texte à trou. Pour des raisons de temps de traitement, cette évaluation est menée sur un échantillon de 150 interviews du corpus *Reticular* (soit environ 5% du corpus total) sélectionnées de manière aléatoire pour une taille d'environ 450 000 mots. Nous ajoutons également son étiquette grammaticale à chaque mot.

Le modèle propose une liste triée par ordre de probabilités de prédictions du mot caché (qui occupe la position centrale du contexte). Par exemple, pour l'énoncé ci-dessous dans lequel le mot « référendum » [4] est masqué, le modèle entraîné avec une fenêtre de 5 considère le contexte restreint qui suit :

Énoncé : « Est-ce que le camp du maintien au sein de l'Union européenne n'est pas en train depuis déjà maintenant des semaines de reproduire les erreurs de 2005 lors du **référendum** constitutionnel, c'est-à-dire au fond en agitant la peur le catastrophisme en permanence ? »
Contexte considéré : ['lors_ADV', 'du_PRP', 'constitutionnel_ADJ', 'c'est-à-dire_ADV']

Avec l'architecture CBOW, le modèle récupère le vecteur du contexte donné (calculé comme le vecteur moyen de tous les mots du contexte) et donne une liste de prédictions de la taille du vocabulaire pour le mot manquant (dont nous restreignons l'affichage aux 5 premières propositions par souci de lisibilité des résultats), triée par ordre de probabilité (i.e. par ordre de similarité la plus élevée entre le vecteur du contexte et le vecteur de la prédiction).

3.3 Évaluation quantitative et qualitative des prédictions

Pour évaluer les prédictions de notre modèle, nous mesurons la moyenne et l'écart-type des distances (notées entre 0 et 1) entre le vecteur des mots cachés et ceux des prédictions faites par le modèle. Les résultats les plus proches de 0 correspondent donc aux cas où le mot à prédire se trouve parmi les 5 premières propositions du modèle.

Le tableau 1 donne les résultats de la prédiction pour chaque modèle avec une distance moyenne par catégorie grammaticale. La deuxième colonne précise la fréquence de chaque catégorie dans le

4. Dans tout l'article, nous utilisons le **gras** pour marquer les mots à prédire dans les énoncés.

corpus d'évaluation. La dernière colonne donne l'écart des moyennes des deux modèles. Les résultats sont affichés sur la base de la troisième colonne, par ordre de distance pour le modèle avec une taille de fenêtre fixée à 5.

Prédictions	Fréquence (en milliers)	Fenêtre de 5		Fenêtre de 15		Variation (en %)
		Distance	Écart-type	Distance	Écart-type	
ADV	41	0.3	0.11	0.31	0.17	3.4 %
PRO	81	0.32	0.08	0.35	0.12	**9.4 %**
KON	26	0.33	0.08	0.35	0.13	4.5 %
DET	42	0.35	0.06	0.35	0.07	**1.4 %**
VER	83	0.35	0.07	0.37	0.08	4.3 %
PRP	56	0.36	0.05	0.38	0.07	4.2 %
NOM	68	0.36	0.08	0.39	0.09	8.3 %
ADJ	21	0.37	0.07	0.4	0.08	**9.6 %**
NUM	5	0.4	0.09	0.46	0.08	**12.3 %**
NAM	20	0.42	0.09	0.45	0.09	7.2 %

TABLE 1 – Distance moyenne des prédictions par catégorie grammaticale

Au premier abord, nous remarquons que les distances moyennes sont toutes, sans exception, plus élevées pour le modèle entraîné avec une taille de fenêtre plus grande et que l'écart-type reste sensiblement le même pour les deux modèles, ce qui laisse entendre que toutes les prédictions de chaque catégorie sont affectées de la même manière par l'augmentation de la taille de la fenêtre.

Concernant la variation des distances moyennes en passant d'un modèle à l'autre, nous remarquons que les prédictions des pronoms, des numéraux, des adjectifs et des noms sont celles qui varient le plus. Ces résultats sont difficilement compréhensibles sans jeter un oeil au comportement des prédictions. Une analyse manuelle des prédictions faites par les deux modèles nous permet de faire les observations suivantes :

1. Concernant la prédiction des numéraux (NUM) :

   — Le modèle entraîné avec une taille de fenêtre plus petite est obligé d'accorder plus de poids aux quelques mots du contexte dont il dispose pour calculer le vecteur le plus proche. Il en résulte que pour la prédiction des numéraux, même s'il ne propose pas forcément le chiffre exact à prédire, il est plus simple pour lui de prédire les suites de chiffres (i.e. le modèle propose des chiffres lorsqu'il y en a déjà dans le contexte, e.g. « **000** » dans « déjà 300 **000** morts »), les dates (e.g. « **11** » dans « attentats du **11** septembre 2001 »), les siècles (e.g. « datent du **XIXème** siècle ») ou les noms de Républiques (e.g. « de la **Ve** République »). En revanche, le contexte lui pose problème lorsqu'il contient des mots polysémiques (e.g. pour le contexte « à peine **10** jours après », le modèle propose des mots comme « emprisonnement », « incarcération » et « perpétuité » à la place du nombre « **10** »).

   — À l'inverse, il est naturellement plus difficile pour le modèle entraîné avec une taille de fenêtre plus grande de prédire les chiffres qui sont souvent utilisés comme des déterminants et nécessitent d'avoir une vision très locale du contexte pour les prédire. La représentation du contexte en sacs de mots empêche toute prédiction lorsque le contexte

contient plus de catégories nominales et verbales. Ainsi, même soumis à évaluation humaine, il est difficile de dire qu'il manque un adjectif numéral dans un exemple comme « il arrive quand même systématiquement en **5ème** position » si on considère les mots indépendamment de leur ordre dans la phrase.

2. Concernant la prédiction des pronoms (PRO) :

   — De la même manière, une fenêtre réduite semble permettre au modèle de reconnaître plus facilement les pronoms personnels et impersonnels (ex.« **il** faut que ») pour lesquels le contexte contient généralement le verbe qu'ils accompagnent, mais l'empêche de prédire les pronoms lorsque le contexte contient les entités nommées auxquelles ils réfèrent. Le modèle montre alors une tendance à proposer plutôt les associations les plus probables pour cette entité (e.g. pour le contexte ['Patrick_NAM', 'Buisson_NAM', 'être_VER', 'quelqu'un_PRO'], le modèle ne propose que des noms de personnes, probablement souvent associées à l'entité nommée « Patrick Buisson » dans le corpus *Wikipédia* au lieu du pronom attendu « **ce** » pour « Patrick Buisson **c'**est quelqu'un de...»).

   — la taille de fenêtre plus grande semble au contraire permettre au modèle entraîné avec une fenêtre de 15 de prédire des mots qui ont un sens proche de celui à prédire, indépendamment de la catégorie grammaticale. Par exemple, pour une phrase « Je n'ai **aucune** espèce d'hésitation », le modèle ne propose que des mots du champ lexical de l'absence (i.e. « pas_ADV », « guère_ADV », « rien_ADV », « aucun_PRO » et « jamais_ADV »).

Nous notons que les catégories des noms communs (NOM), adjectifs (ADJ) et noms propres (NAM) présentent les distances moyennes les plus élevées. Ces résultats peuvent s'expliquer en partie par le fait qu'il s'agit de catégories productives et en liste ouverte, qui présentent parfois des formes très peu fréquentes dans le corpus d'entraînement. Puisque nous entraînons notre modèle sur un corpus très différent de notre corpus d'évaluation, nous pouvons également nous questionner sur l'impact des représentations apprises sur la prédictibilité des noms. En effet, il est peu probable que les catégories qui relèvent plutôt de la syntaxe (comme les conjonctions ou déterminants) présentent un usage très différent d'un corpus à l'autre. En revanche, les catégories ouvertes comme les noms et adjectifs ont plus de probabilité de voir leur sens modifié en passant d'un corpus encyclopédique à un corpus politique plus subjectif, rendant leur prédiction exacte plus compliquée.

En observant les étiquettes grammaticales des mots prédits par les modèles, nous pouvons observer néanmoins qu'elles sont également les catégories les plus faciles à reconnaître. Le tableau 2 donne le ratio moyen, par catégorie grammaticale, de prédictions d'étiquette grammaticale identiques à celle du mot à prédire parmi les 5 premières prédictions données par chaque modèle.

En effet, les résultats pour les noms communs et propres, les verbes, les adverbes et les adjectifs sont les plus élevés, ce qui signifie qu'il est plus facile pour les modèles de prédire si le mot manquant est un nom plutôt qu'un déterminant. Là encore, la variation quand on passe d'un modèle à l'autre est intéressante, puisque le modèle entraîné avec une plus grande fenêtre montre une tendance plus élevée à donner des prédictions d'étiquettes différentes du mot à prédire, particulièrement lorsqu'il s'agit de noms.

En réalité, en observant les données, nous pouvons nous rendre compte que le modèle, au lieu de ne proposer que des noms qui lui paraissent proches des mots qui apparaissent dans le contexte étudié, propose également les verbes et adjectifs dont le sens est proche du mot à prédire. Par exemple,

Prédictions	Fenêtre de 5 (en %)	Fenêtre de 15 (en %)
NOM	48	35
VER	41	40
ADV	32	27
NAM	23	12
PRO	18	18
ADJ	17	9
NUM	14	4
KON	6	7
PRP	3	1
DET	0	0

TABLE 2 – Ratio moyen de prédiction des catégories grammaticales

pour une phrase « il n'est pas en situation d'être candidat à l'**élection** présidentielle », le modèle va prédire des mots comme « présidentiel_ADJ » et « réélire_VER » en plus de « élection_NOM ». Ce phénomène est encore une fois dû à la prise en compte du contexte comme un sac de mots, qui est corrigé lorsque la fenêtre prise en compte est plus petite puisque le modèle fait un focus sur les cooccurrents directs du mots (ici l'article défini « le » et l'adjectif « présidentielle »). Néanmoins, dans certains contextes où les éléments nécessaires à la prédiction du mot sont plus éloignés dudit mot, cette vision plus élargie du contexte peut s'avérer utile. C'est le cas par exemple pour l'étude de la phrase « À l'origine, certaines personnalités chez vous militaient pour une large primaire à **gauche** qui engloberait tout le monde, des socialistes jusqu'à Jean-Luc Mélenchon. » pour laquelle le modèle donne des prédictions comme « gauche », « écologiste », « communiste » et « socialiste ».

De manière générale, il semble effectivement que réduire la taille du contexte pour l'apprentissage permet de mieux intégrer la syntaxe dans les prédictions du modèle, mais une taille de contexte plus grande est parfois nécessaire lorsque les éléments indispensables à la compréhension de l'extrait se trouvent plus loin ou que les éléments du contexte sont trop ambigus.

Si les objectifs du modèle sont de détecter la nomination, il est peu probable que tous les éléments nécessaires à la compréhension du nom soient disponibles dans le contexte restreint autour du mot-cible. Cette expérience nous conforte donc dans l'idée qu'un contexte plus large est préférable pour l'apprentissage de nos modèles.

En renouvelant l'expérience, cette fois avec un modèle entraîné avec une taille de fenêtre de 10, nous nous apercevons que les distances moyennes et écarts-types varient très peu par rapport à ceux mentionnés dans le tableau 1 (i.e. seulement 0.01 ou 0.02 points d'écart, toutes catégories confondues avec le modèle entraîné avec un contexte fixé à 15). Comme pour l'expérience menée par (Patel & Bhattacharyya, 2017), il semble donc qu'il existe un seuil de fenêtre à partir duquel les résultats ne varient plus. Cette observation nous pousse à fixer notre taille de contexte à 10 pour nos prochaines expérimentations.

4 Prédiction de mot-pivot

En utilisant le même principe que dans notre expérience sur les catégories grammaticales présentée dans les deux sections précédentes, nous souhaitons observer le comportement de notre modèle pour

la prédiction d'un mot pivot selon la méthode habituelle de l'analyse du discours. Nous supposons que si le modèle propose dans sa liste de prédictions des mots très éloignés du mot à prédire, ce résultat implique que le mot en question est employé de manière inhabituelle (i.e. apparaît dans un contexte peu probable) et peut relever de la nomination.

Pour cette étude, nous réunissons un sous-corpus autour du mot-pivot « Europe », déjà identifié par d'autres travaux (Gauthier, 2016) comme candidat nomination et très fréquent dans notre corpus d'interviews. Nous extrayons grâce au concordancier de l'outil TXM (Heiden *et al.*, 2010) 5855 contextes contenant la forme « Europe ».

Pour les raisons citées en fin de section précédente, nous travaillons cette fois avec un modèle entraîné avec une taille de fenêtre de 10. L'analyse manuelle des résultats nous permet déjà d'identifier plusieurs sens pour le mot « Europe », à savoir :

— un **territoire** : le modèle parvient à identifier les usages du mot au sens de lieu lorsque des mots qui s'en rapprochent ou qui indiquent une position font partie du contexte (e.g. « Monde », « France », « dans » , « en »).

— une **puissance économique** : on retrouve pour plusieurs contextes des prédictions du champ lexical de la puissance économique (e.g. « empire », « compétitivité », « prospérité », « capitalisme », « mondialisation », « libéralisation »). Des prédictions comme « utopie » ou « idéologie » permettent déjà de mettre en lumière cette vision colorée de l'entité dans un contexte comme « l'avenir progressiste de l'**Europe** et du monde ».

— un **peuple** : on trouve dans certains cas comme « l'humanité, notre pays, l'**Europe**, le monde » ou « l'**Europe** immobile, l'Europe du sceptique » des prédictions comme « civilisation » ou « civiliser » qui présentent l'Europe comme une entité dotée de vie et d'intelligence.

— une **victime** ou un **bourreau** : d'autres exemples permettent de déceler l'inquiétude du locuteur pour ou sur l'entité, par exemple dans « l'**Europe** cause de tout le mal », contexte pour lequel le modèle propose des mots comme « esclavage », « humiliation », « trahison » et « injustice », ou au contraire dans l'exemple « c'est une menace pour l'Europe » pour lequel il propose des mots comme « victime », « trahison » ou encore « esclave ».

Néanmoins, ces résultats restent difficiles à exploiter puisque la majorité des prédictions comprennent des résultats très éparses, souvent très éloignés sans que l'on puisse déterminer s'ils sont la conséquence d'une nouvelle utilisation du mot ou d'une mauvaise représentativité des données dans le corpus d'entraînement.

5 Discrimination de sens

Dans une seconde approche, nous envisageons de confronter directement des modèles appris sur différents corpus (chacun représentatif d'un locuteur différent) pour mesurer la variation sémantique interdiscursive (i.e. les changements de sens appliqués à chaque mot en fonction du locuteur).

Nous souhaitons pour cela comparer un modèle appris sur le corpus *Wikipédia* avec différents modèles appris sur des concaténations du corpus *Wikipédia* et les sous-corpus contenant à chaque fois les interviews d'un seul candidat. En suivant la méthode utilisée par Pierrejean & Tanguy (2018), nous comptons ainsi faire une comparaison par binômes de modèles en observant la variation des voisins distributionnels les plus proches (*Nearest Neighbors*) de chaque mot (i.e les mots dont le score de

similarité cosinus est le plus élevé par rapport au mot cible). Après avoir identifié le vocabulaire commun à chaque paire de corpus (i.e. en retirant de l'étude tous les mots du corpus *Wikipédia* qui n'apparaissent pas dans les interviews), nous utiliserons la distance de Jaccard pour mesurer le taux de variation entre chaque modèle.

Nous souhaitons également utiliser ces représentations pour regrouper des clusters de sens sur le modèle de Pelevina *et al.* (2017) pour tenter d'identifier de nouveaux aspects sémantiques, mais également rapprocher les usages de mots identiques d'un locuteur à l'autre.

Grâce à cette étude, nous souhaitons mettre au jour les représentations qui évoluent d'un modèle à l'autre, pour identifier les usages qui relèveraient soit d'un usage spécifique au locuteur, soit d'un usage spécifique au genre de l'interview politique.

Cette étude devra néanmoins prendre en compte l'instabilité inhérente aux méthodes d'apprentissage des modèles (Pierrejean & Tanguy, 2019) dans l'évaluation des représentations de variations sémantiques inter-locuteur pour discriminer les variations dues à l'entraînement des modèles des variations effectivement dues à des usages différents.

6 Conclusion et perspectives

Dans l'objectif de fournir une méthode capable de rendre compte de la variation sémantique de la nomination, qui viendrait compléter les fonctionnalités des outils déjà utilisés par l'analyse du discours, cet article propose une exploration de méthodes distributionnelles pour repérer de manière automatique des candidats nominations.

Notre problématique, qui nécessite de détecter les usages particuliers des noms (i.e. dont le sens s'écarte de l'habituel), se heurte à la représentation vectorielle unique de chaque forme calculée par les modèles prédictifs. Pour répondre à cette question, cet article propose un état de l'art des méthodes distributionnelles appliquées à la désambiguïsation lexicale et la représentation de la variété sémantique sous forme de vecteurs de sens duquel nous nous inspirons pour nos propres travaux.

Nos expérimentations, menées sur un corpus de transcriptions d'interviews politiques enregistrées dans des émissions de radio, questionnent la capacité de modèles prédictifs à représenter la variation sémantique d'un discours à l'autre. Une première expérience, menée sur la prédictibilité des catégories grammaticales, nous permet d'évaluer l'impact du contexte pris en compte dans l'entraînement des modèles sur la représentation des mots porteurs de sens. Sur le modèle de travaux en analyse de discours, nous utilisons le modèle qui résulte de cette expérimentation pour observer son utilisabilité pour la prédiction de nouveaux usages d'un mot-pivot. Les résultats, bien qu'encourageants, restent difficiles à exploiter en l'état et ne permettent pas de discerner facilement les résultats effectivement dus à un usage inhabituel ou à une lacune du modèle appris. Enfin, en nous inspirant d'approches concentrées sur l'étude des voisins distributionnels, notre article amorce une réflexion sur une méthode de détection de la variation sémantique interdiscursive basée sur une comparaison par binômes de modèles représentatifs respectivement d'un usage encyclopédique *vs.* en discours du lexique.

Nous espérons, à terme, que nos travaux nous permettrons de définir des critères de sélection automatique de candidats nominations à intégrer à des outils déjà existants pour l'analyse de discours.

Pour des raisons pratiques, nous avons décidé de ne pas nous pencher plus sur l'aspect diachronique de l'étude de la nomination, qui est normalement repérable en premier lieu sur le même modèle que la dénomination de Kleiber (1984), par son acte de baptême (Siblot, 2001) (i.e. lorsque le locuteur décide de nommer par tel nom précis l'entité de son choix). Ce phénomène n'étant repérable qu'à

l'introduction de la nomination, nous pouvons douter de la représentativité de notre corpus sur cet aspect, et nous cantonner pour le moment à une approche synchronique nous paraît plus sage. Néanmoins, nous ne tirons pas de trait définitif sur la perspective d'étendre ultérieurement l'approche à d'autres corpus pour inclure cette particularité à notre travail.

Aussi, nous n'abordons pas dans ce papier la question de modèles plus performants largement exploités dans le domaine du TAL, tels que BERT (Devlin *et al.*, 2018) et ELMo (Peters *et al.*, 2018). Nous avons choisi de débuter nos travaux avec des modèles dont la documentation nous paraît plus accessible et l'entraînement moins coûteux pour éprouver nos méthodes, mais une expérimentation de modèles neuronaux profonds est prévue dans la poursuite de notre étude.

Références

BARONI M., DINU G. & KRUSZEWSKI G. (2014). Don't count, predict ! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, p. 238–247 : Association for Computational Linguistics. DOI : 10.3115/v1/P14-1023.

BLOEM J., FOKKENS A. & HERBELOT A. (2019). Evaluating the consistency of word embeddings from small data. *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*.

CALABRESE L. (2018). Faut-il dire migrant ou réfugié ? débat lexico-sémantique autour d'un problème public. *Langages*, **210**, p.105–124. DOI : 10.3917/lang.210.0105.

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*.

DONGSUK O., SUNJAE K., KYUNGSUN K. & YOUNGJOONG K. (2018). Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph. In *Proceedings of the 27th International Conference on Computational Linguistics* : Association for Computational Linguistics.

FIRTH J. (1957). *A synopsis of linguistic theory*. Blackwell, Oxford.

GAUTHIER G. (2016). Le « printemps érable » au québec : « grève » ou « boycott » ? les enjeux stratégiques d'un conflit de nomination. *Argumentation et Analyse du Discours (AAD)*, **17**. DOI : 10.4000/aad.2248.

HARRIS Z. (1954). Distributional structure. *Word*, **10**, 146–162.

HEIDEN S., MAGUÉ J.-P. & PINCEMIN B. (2010). Txm : Une plateforme logicielle open-source pour la textométrie – conception et développement. In *Proc. of 10th International Conference on the Statistical Analysis of Textual Data - JADT 2010 (Volume 2)*, p. p.1021–1032, Roma, Italy.

KLEIBER G. (1984). « dénomination et relations dénominatives ». *Langages*, **76**, p.77–94. DOI : 10.3406/lgge.1984.1496.

KOREN R. (2016). La nomination et ses enjeux socio-politiques : Introduction. *Argumentation et Analyse du Discours (AAD)*, **17**. DOI : 10.4000/aad.2295.

LEBART L. & SALEM A. (1994). *Statistique textuelle*. Dunod.

LEVY O. & GOLDBERG Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, p. 302–308 : Association for Computational Linguistics. DOI : 10.3115/v1/P14-2050.

LEVY O., GOLDBERG Y. & DAGAN I. (2015). Improving distributional similarity with lessons learned from word embeddings. In *Transactions of the Association of Computational Linguistics, Volume 3*, p. 211–225 : Association for Computational Linguistics. DOI : 10.1162/tacl_a_00134.

LONGHI J. (2017). Humanités, numérique : des corpus au sens, du sens aux corpus. *Questions de communication*, **31**, p.7–17.

MARCHAND P. & RATINAUD P. (2012). L'analyse de similitude appliquée aux corpus textuels : les primaires socialistes pour l'élection présidentielle française (septembre-octobre 2011). In *Actes des 11eme Journées internationales d'Analyse statistique des Données Textuelles. JADT*, p. 687–699.

MAZIÈRE F., Éd. (2018). *L'analyse du discours : Histoire et pratiques. "Que sais-je ?"*. Presses Universitaires de France.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint, arXiv : abs/1301.3781*.

NOUVEL D., EHRMANN M. & ROSSET S. (2015). *Les entités nommées pour le traitement automatique des langues*. ISTE Group.

PATEL K. & BHATTACHARYYA P. (2017). Towards lower bounds on number of dimensions for word embeddings. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*.

PELEVINA M., AREFYEV N., BIEMANN C. & PANCHENKO A. (2017). Making sense of word embeddings. *arXiv preprint, arXiv : abs/1708.03390v1*.

PENGAM M. & JACKIEWICZ A. (2019). Sens et emplois de l'expression « musulmans modérés » dans les discours médiatiques. *Open Library of Humanities*, **5**, p.45. DOI : 10.16995/olh.431.

PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. *arXiv preprint arXiv :1802.05365v2*.

PIERREJEAN B. & TANGUY L. (2018). Towards qualitative word embeddings evaluation : Measuring neighbors variation. In *Conference of the North American Chapter of the Association for Computational Linguistics : Student Research Workshop*, p. 32–39, New-Orleans, United States. HAL : hal-01806468.

PIERREJEAN B. & TANGUY L. (2019). Investigating the stability of concrete nouns in word embeddings. In *13th International Conference on Computational Semantics*, p. 32–39, Gothenburg, Sweden. HAL : hal-02073705.

REINERT M. (1993). Les "mondes lexicaux" et leur "logique" à travers l'analyse statistique d'un corpus de récits de cauchemars. *Langage et société*, **66**, p.5–39.

RUAS T., GROSKY W. & AIZAWA A. (2019). Multi-sense embeddings through a word sense disambiguation process. *Expert Systems with Applications*, **136**, 288–303. DOI : 10.1016/j.eswa.2019.06.026.

SCHMID H. (2013). Probabilistic part-of-speech tagging using decision trees. *New methods in language processing*, p. 154.

SIBLOT P. (2001). De la dénomination à la nomination. *Cahiers de praxématique*, **36**, p.189–214. DOI : 10.3406/lgge.1997.2124.

TOURNIER M. (1981). Spécificité politique et spécificité lexicale. *Mots. Les langages du politique*, **2**, 5–10.

VIAL L., LECOUTEUX B. & SCHWAB D. (2017). Sense embeddings in knowledge-based word sense disambiguation. In *12th International Conference on Computational Semantics*. HAL : hal-01599685.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

L'adaptabilité comme compétence pour les systèmes de dialogue orientés tâche

Oralie Cattan
Université Paris-Saclay, LIMSI, Campus Universitaire bâtiment 507, Rue du Belvédère, 91400 Orsay, France
prénom.nom@limsi.fr
Qwant Research, 7 Rue Spontini, 75116, France
initiale.nom@qwant.com

RÉSUMÉ

Étendre les capacités d'adaptabilité des systèmes à toujours plus de nouveaux domaines sans données de référence constitue une pierre d'achoppement de taille. Prendre en charge plus de contenus serviciels constitue un moyen de diversifier l'éventail des capacités de compréhension des systèmes de dialogue et apporterait un véritable intérêt pour les utilisateurs par la richesse des échanges qu'elle rendrait possibles. Pour favoriser les progrès dans ce sens, la huitième édition du défi Dialog State Tracking Challenge introduit des pistes exploratoires permettant d'évaluer les capacités de généralisation et d'habileté des systèmes à composer à la fois avec la nouveauté et avec plusieurs domaines de tâches complexes. L'objectif de cet article est de rendre compte des recherches du domaine et contribue à donner des éléments de réponse de manière à mieux comprendre les limites des systèmes actuels et les méthodes appropriées pour aborder ces défis.

ABSTRACT

Adaptability as a skill for goal-oriented dialog systems

Extending the adaptability of systems to new domains without reference data is a major stumbling block. Taking charge of more service contents constitutes a means of diversifying the range of capacities in understanding for dialogue systems bringing real interest to users through the wealth of exchanges it would make possible. To promote progress in this direction, the eighth edition of the Dialog State Tracking Challenge introduces exploratory tracks allowing to assess the general capacities and the abilities of systems to deal with both novelty and multiple complex task domains. The objective of this paper is to report on research in the field and help to provide answers so as to better understand the limits of current systems and the appropriate methods to tackle these challenges.

MOTS-CLÉS : système de dialogue, suivi de l'état du dialogue guidé par des schémas, adaptabilité.

KEYWORDS: dialogue system, schema-guided dialog state tracking, adaptability.

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 85–95.
HAL : hal-02786190.

1 Introduction

Les systèmes de dialogue sont des systèmes d'interface homme-machine, qui par le biais de différents canaux de communication, qu'ils soient textuels, vocaux ou visuels, voire multimodaux, permettent d'offrir des services en mettant en correspondance un utilisateur humain et un système informatique.

Il existe principalement deux grands types de systèmes de dialogue. Les systèmes orientés tâche qui interagissent avec les utilisateurs pour accomplir des tâches spécifiques. Cela peut aller de tâches simples et bien définies comme programmer une alarme à une heure précise à des tâches complexes comme la planification de voyages ou la négociation de contrats. À l'inverse, les systèmes non orientés tâche engagent généralement les utilisateurs dans des conversations brèves qui ne nécessitent pas nécessairement de tâches à accomplir et dont le but est de divertir.

En fonction du type de conversation, la structure des dialogues et les objectifs de compréhension varient et naturellement, une conversation peut impliquer un mélange d'interactions orientées tâche et non orientées tâche. Ainsi, dans la réalité, il peut exister un certain chevauchement entre ces deux types de systèmes.

Les systèmes de dialogue orientés tâche suivent généralement une architecture modulaire présentée dans la figure 1, composée a minima, d'un enchaînement de trois composantes.

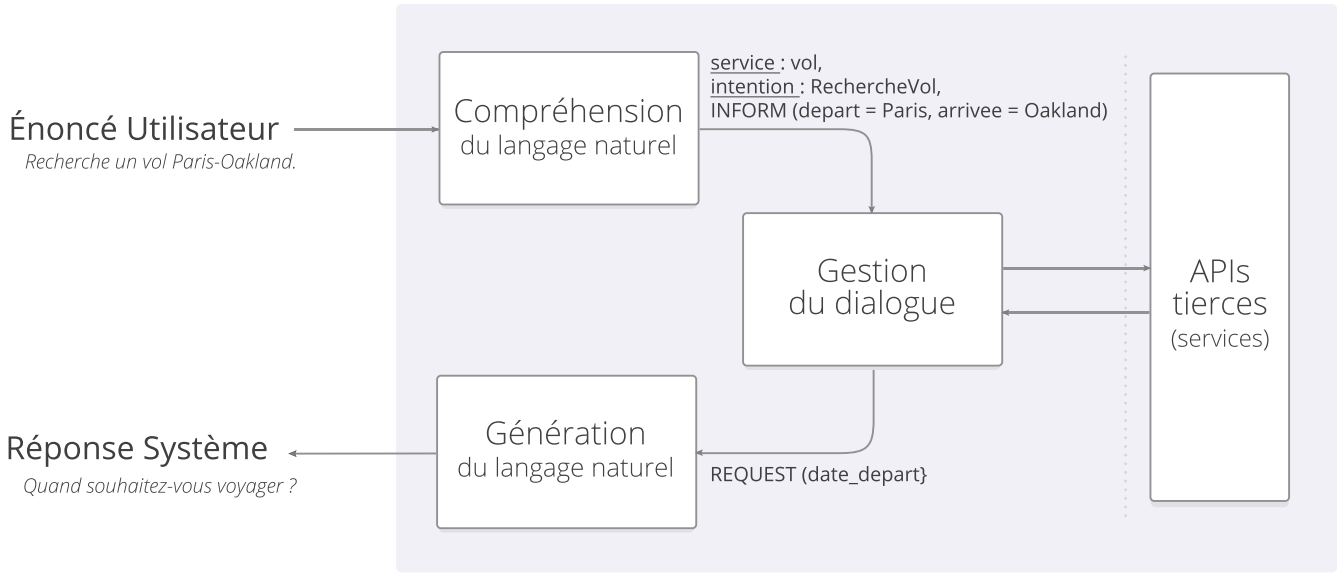


FIGURE 1 – Architecture typique d'un système de dialogue.

La compréhension du langage naturel (*natural language understanding* en anglais) fait référence à la tâche qui consiste à extraire des éléments de sens véhiculés par la requête de l'utilisateur pour construire une représentation symbolique manipulable par le système.

Elle englobe traditionnellement trois tâches :

— L'identification du domaine. Dans le cas des systèmes de dialogue multidomaines, le domaine correspond à la tâche e.g. la gestion des alarmes, la réservation de vols, etc.. Elle se fonde sur une ontologie qui permet de spécifier un ensemble d'informations (intentions et concepts) nécessaires pour modéliser les connaissances du domaine.

— L'identification de l'intention. Elle correspond à une opération de classification permettant de déterminer l'objectif de l'utilisateur en fonction du domaine préalablement identifié.

— La reconnaissance des concepts (*slots* en anglais). Elle correspond à une opération d'étiquetage des segments pertinents de l'énoncé utilisateur du point de vue de l'intention. Détecter ces concepts et extraire ses valeurs permettront un traitement particulier (interaction avec une source de connaissances, stockage, etc.).

La gestion du dialogue (*dialogue management* en anglais) fait référence au module qui s'occupe de la conduite du dialogue dans le but d'accomplir une tâche. Cette gestion de l'échange d'information entre l'utilisateur et le système va de la prise de décision ou action aux moyens de faire progresser l'interaction. Pour ce faire, le gestionnaire de dialogue met à jour son état interne grâce notamment au contenu sémantique obtenu en sortie du module de compréhension et à l'historique des tours précédents qui sont nécessaires au suivi de l'avancement de la tâche de dialogue. C'est aussi à ce niveau qu'un accès à des sources de connaissances (bases de données, interfaces de programmation d'application ou APIs, etc.) peut être réalisé dans le but récupérer les données utiles à la résolution du besoin en information de l'utilisateur.

Enfin, la génération de langage naturel (*natural language generation* en anglais) fait référence à la tâche qui consiste à produire la réponse du système en langue naturelle sur la base du contenu fourni par le module précédent.

Que ces composants soient entraînés individuellement ou de bout en bout, les modèles résultants sont toujours développés pour une tâche spécifique ou un domaine particulier. Cette conception implique des limites fortes tant au niveau de leurs mises à jour qu'au niveau de leurs extensions à de nouveaux domaines où le manque de données fait effet de goulot d'étranglement. De plus, si les systèmes de dialogue orientés tâche sont largement adoptés dans l'industrie (*Siri*, *Cortana*, *Alexa*, etc.) avec une grande fiabilité, leur robustesse et portabilité multilingue et multidomaine restent limitées. En prenant en charge un nombre croissant de services par le biais d'APIs tierces pour fournir des contenus ou pour intégrer des périphériques externes, ces systèmes sont confrontés à un plus fort degré de chevauchement inter- et trans- domaines, ce qui entrave considérablement le développement rapide de nouvelles compétences.

Depuis quelques années, il existe un fort intérêt en ce qui concerne la conception de systèmes agnostiques au domaine d'application bénéficiant d'un apprentissage moins gourmand en données et couvrant une multitude de domaines pour atteindre des objectifs plus complexes.

Dans la suite de l'article, nous présentons les problèmes posés par l'adaptabilité des systèmes en caractérisant ses approches ainsi que les ressources disponibles pour conclure sur les perspectives.

2 L'adaptabilité comme compétence

Tout au long de leurs vies, les êtres humains démontrent d'une capacité d'apprendre à apprendre (Thrun & Pratt, 1998). Un enfant qui apprend la marche généralisera rapidement cette nouvelle compétence à d'autres contextes (présence de dénivelés, de bourrasques ou d'obstacles, etc.) avec un minimum de temps d'adaptation en s'appuyant sur son expérience et sur ses observations antérieures. Les méthodes d'apprentissage automatique traditionnelles ne rendent pas compte de cette capacité à généraliser et à mobiliser les connaissances acquises pour faire face à une tâche différente de celle rencontrée lors de l'entraînement.

Dans une entreprise réelle où les systèmes sont développés rapidement et devraient fonctionner de manière robuste pour une variété croissante de domaines et tâches, un apprentissage rapide, continu et efficace à partir d'un nombre limité d'exemples devient indispensable. C'est pourquoi, dans les sections suivantes, nous faisons un tour d'horizon des techniques actuelles qui permettent dans une certaine mesure de prendre en considération ces aspects.

2.1 Vers des systèmes adaptables

Nous sommes devenus capables d'entraîner des réseaux de neurones profonds pour apprendre à relier des entrées à des sorties souhaitées à partir de grandes quantités de données.

L'idée de transférer les connaissances et les compétences acquises en exploitant des données existantes, non nécessairement directement liées à la tâche pour apprendre une autre tâche à partir d'autres données n'est pas nouvelle (Caruana, 1993). On retrouve aujourd'hui cette idée, sous le nom d'apprentissage par transfert (*transfert learning* en anglais), appliquée pour entraîner un réseau sur une tâche à partir d'un modèle déjà entraîné sur une tâche similaire. En traitement automatique des langues par exemple, Zoph *et al.* (2016) ont exploité l'apprentissage par transfert pour une tâche de traduction automatique pour pallier l'absence de corpus annotés dans le cas de langues peu dotées, en tirant parti des données abondantes disponibles dans d'autres langues.

Une classe de problèmes d'apprentissage automatique qui s'apparente à celle de l'être humain qui apprend de ses erreurs en s'adaptant à son environnement, connue sous le nom d'apprentissage par renforcement (*reinforcement learning* en anglais) a permis l'émergence de systèmes performants dans des domaines impliquant une gestion de séquences d'action tels que le jeu, la robotique ou le dialogue. Toutefois, si ce type d'apprentissage a permis d'éviter une modélisation de la décision *ad hoc* du problème considéré, des connaissances expertes du domaine restent nécessaires.

Dans une tout autre perspective, des approches ont été présentées récemment sur la manière de développer des systèmes, des capacités d'apprentissage tout au long de la vie avec l'apprentissage continu (*continual learning* en anglais) (Parisi *et al.*, 2019). Jusqu'alors, une fois la phase d'entraînement achevée, les modèles demeuraient statiques : les structures et les poids entre les neurones étaient fixés. Ce mode d'apprentissage permet d'obtenir des modèles pouvant être améliorés en permanence en accumulant continuellement de nouvelles connaissances sur différentes tâches et convient à des applications où la base d'entraînement évolue dans le temps. Pour autant, le gain de flexibilité inhérent à cette capacité d'adaptation reste limité aux domaines de tâche définis et s'obtient au détriment d'une implémentation plus complexe du système.

Finalement, l'un des aspects les plus frappants de l'apprentissage humain est l'aptitude à apprendre de nouveaux concepts à partir d'un nombre limité d'exemples. Cette capacité contraste fortement avec les méthodes traditionnelles d'apprentissage automatique, qui nécessitent une quantité de données importante. Ces dernières années, on a vu apparaître une réflexion en méta sur l'apprentissage et des propositions d'algorithmes capables de résoudre le problème d'apprentissage en zéro ou quelques coups (en anglais, *zero-shot learning* et *few-shot learning*), autrement dit, avec zéro ou peu de données annotées. Introduit dans le domaine de la vision par ordinateur pour la reconnaissance d'objets (Larochelle *et al.*, 2008; Palatucci *et al.*, 2009), le méta-apprentissage (*meta-learning* en anglais) (Vanschoren, 2018; Wang *et al.*, 2019) a permis de méta-entraîner directement des modèles ayant une bonne performance en généralisation pour de nouvelles classes d'objets ou de nouveaux domaines de tâches à partir de peu de données.

2.2 Le cas des systèmes de dialogues

Comme on l'a vu dans la section 1, l'approche prédominante dans le domaine de la conception des systèmes de dialogue orientés tâche suit un découpage modulaire qui a pour principal avantage de permettre d'évaluer indépendamment les performances de chaque module en offrant la possibilité d'analyser l'origine des erreurs. Cependant, elle a pour défauts d'impliquer un processus d'annotation lourd et d'être peu générique, car les modèles doivent être mis à jour dès que l'on souhaite porter le système à une nouvelle tâche.

Le travail de Tur *et al.* (2014) et celui de Ferreira *et al.* (2015) pour le module de compréhension a permis de limiter ce besoin en données annotées en utilisant un apprentissage sans données de référence. On retrouve aussi cette forme d'apprentissage appliquée pour prendre en compte de nouveaux domaines (Kumar *et al.*, 2017), de nouvelles intentions (Chen *et al.*, 2016; Xia *et al.*, 2018) et de nouveaux concepts (Bapna *et al.*, 2017; Lee & Jha, 2019). C'est dans cette même perspective que Lin & Xu (2019) ont proposé une approche se fondant sur la détection de nouveautés (en anglais *novelty detection*) pour identifier des intentions non vues dans la base d'entraînement.

En ce qui concerne le module de génération, Wen *et al.* (2016) ont proposé une technique d'augmentation artificielle des données pour générer des exemples à partir d'ensembles de données délexicalisées hors domaines. Un modèle appris à partir de ces données synthétiques et ajusté sur un ensemble plus petit de données du domaine d'intérêt a permis une généralisation à de nouveaux domaines. Récemment, Mi *et al.* (2019) ont porté l'algorithme MAML (Finn, 2018), introduit pour l'interaction robotique à la génération du langage. En méta-entraînant un modèle afin qu'il soit capable de généraliser à des domaines absents de l'ensemble d'entraînement avec seulement une petite quantité de données, les auteurs ont pu comparer les performances de ce modèle, dans le cas où les domaines étaient connexes ou distants.

Développées dans le cadre d'applications multidomaines sans que de nouveaux paramétrages ou apprentissages spécifiques soient nécessaires et avec un besoin en données moindre, ces approches permettent d'explorer de nouvelles façons de réaliser des systèmes de dialogue agnostiques au domaine de tâches, indépendants de toute autre technique de détection de nouveauté. Bien que très prometteuses, elles obtiennent des résultats sur des ensembles de données synthétiques et nécessiteraient davantage de tests sur des données réelles. Aussi, ces recherches ne se concentrent que sur l'adaptation de modules individuels d'un système de dialogue alors que plusieurs résultats expérimentaux ont montré que des améliorations peuvent être apportées par une résolution jointe de certaines tâches, notamment l'identification de l'intention et la reconnaissance des concepts (Xu & Sarikaya, 2013; Hakkani-Tür *et al.*, 2016; Zhang & Wang, 2016; Liu & Lane, 2016).

3 Les ressources disponibles

Les ensembles de données existants permettant d'entraîner des systèmes de dialogue orientés tâche tels que DSTC2 (Henderson *et al.*, 2014) et Multi-Domain Wizard-of-Oz (MultiWOZ) (Budzianowski *et al.*, 2018) se fondent sur une ontologie statique par domaine avec un nombre de domaines, d'intentions et de concepts fixe. Comparé au corpus DSTC2 ne contenant qu'un domaine de tâche, MultiWOZ a été introduit dans une volonté d'augmenter la portabilité multidomaine des systèmes avec 7 domaines et un millier de dialogues étiquetés par domaine (cf. Table 3).

	N° de dialogues	N° de domaines	N° de concepts	N° de valeurs
DSTC2	1 612	1	8	212
MultiWOz	8 438	7	24	4 510
SGD	18 624	16	214	14 139
MetalWoz	40 388	47	-	-

TABLE 1 – Principales caractéristiques des ensembles de données.

Parce que ces derniers ne capturent pas les défis susmentionnés, deux corpus à grande échelle ont récemment été rendus publics : Meta-Learning Wizard of Oz (MetaLWOz) (Shalyminov *et al.*, 2020) et Schema-Guided Dialogue (SGD) Rastogi *et al.* (2019).

3.1 Meta-Learning Wizard of Oz

Couvrant près de 50 domaines avec un total de 227 tâches, l'ensemble de données MetaLWOz se compose de 40 000 dialogues humain-humain. Les dialogues ont été collectés suivant un protocole de type Magicien d'Oz (pour l'anglais *Wizard-of-Oz*) (Kelley, 1984) dans lequel les participants se voient attribuer un domaine d'intérêt et une tâche spécifique qu'ils sont invités à poursuivre via le dialogue. Aucune API ou base de connaissances de domaine n'est disponible et les participants utilisent librement des concepts fictifs tout en devant rester cohérents. MetaLWOz est spécifiquement conçu pour réduire la quantité de données nécessaires pour adapter des systèmes de dialogue à de nouveaux domaines et permettre le développement de modèles génératifs reposant sur le paradigme du méta-apprentissage. Aucune annotation n'est fournie, hormis des descriptions en langue naturelle des domaines et des intentions.

3.2 Schema-Guided Dialogue State Tracking

Introduit à l'occasion de la 8e édition du Dialog State Tracking Challenge (Kim *et al.*, 2019), SGD (Rastogi *et al.*, 2019) comprend plus de 18 000 dialogues et couvre 16 domaines. À l'instar de MetaLWOz, il s'agit de tester le pouvoir de généralisation des modèles à des domaines non vus dans la base d'entraînement. En outre, plusieurs services peuvent être associés à un domaine, ce qui entraîne des problèmes de chevauchement inter- et trans- domaines, comme on le verra dans la section suivante. Bien que proposé pour évaluer la tâche de suivi de l'état, SGD peut aussi servir de banc d'essai pour évaluer individuellement les différents composants d'un système de dialogue. En effet, les annotations (classes et descriptions en langue naturelle) des domaines, des services, des intentions et des concepts sont fournies.

3.3 Les difficultés

La piste 4 de la 8e édition du Dialog State Tracking Challenge intitulé Schema-Guided Dialogue State Tracking est un défi de recherche dédié à l'évaluation du suivi de l'état du dialogue dans un cadre pratique, celui d'un système de dialogue confronté aux défis :

— d'évolutivité : la prise en charge d'un grand nombre et d'une grande variété de services à travers l'utilisation d'APIs ne doit pas entraîner une dégradation des performances du système ou remettre en cause sa structure ;

— de réutilisabilité : la gestion des données doit être efficace dans le cas où plusieurs services partagent des éléments communs ;
— d'extensibilité : l'ajout de nouvelles APIs entraîne un certain coût de développement et de maintenance qu'il serait souhaitable de limiter.

Pour un cas concret, une demande d'information comme *recherche un vol Paris-Oakland* déclenche par le système la recherche de services de réservation de vols en mesure de considérer les critères de l'utilisateur sur la base des informations présents dans l'énoncé. Les systèmes de dialogue orientés tâche nécessitent des représentations d'état explicites pour notamment interagir avec des sources externes (ici, des APIs tierces). Cependant, plusieurs services peuvent correspondre et les APIs imposent des contraintes d'accès définies par le fournisseur du service et donc des interfaces qui peuvent être différentes. C'est pour cette raison qu'un registre des schémas d'utilisation des APIs accompagne l'ensemble de données SGD. Chaque schéma définit les fonctions et paramètres d'accès attendus par une API ainsi qu'une description en langue naturelle pour le service, mais aussi pour les intentions et les concepts qui s'y rapportent.

La figure 2 illustre la manière dont les annotations correspondants à deux schémas de services similaires de réservation de vols conditionnent les états et créent des chevauchements. D'un côté l'intention *RechercheVol* et le concept *depart* du service A ainsi que l'intention *TrouveVol* et le concept *origine* du service B se correspondent. De l'autre l'intention *ReserveVol* et le concept *date_retour* se retrouvent dans les deux services.

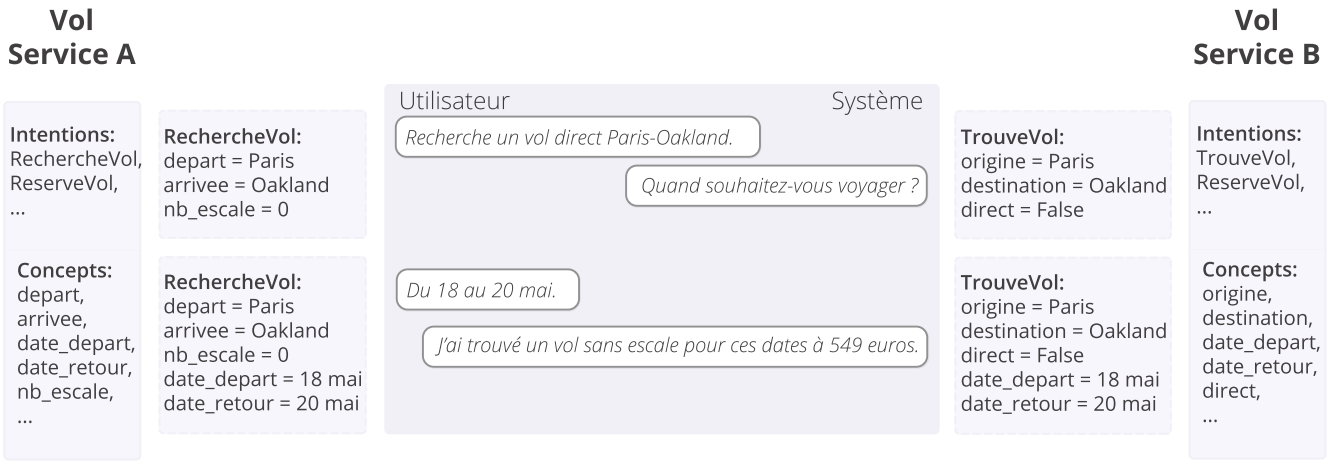


FIGURE 2 – États du dialogue conditionnés par les schémas correspondant à deux services similaires de réservations de vols. Figure adaptée en français de Rastogi *et al.* (2019).

Hormis ces problèmes de chevauchement, les systèmes développés doivent faire face à un cas particulier d'apprentissage où domaines, services, intentions et concepts peuvent ne pas être présents dans l'ensemble des exemples du corpus d'apprentissage. Par exemple, les domaines *Voyage* et *Météo* ne sont présents que dans l'ensemble de développement et le domaine *Alarme* n'est présent que dans l'ensemble de test.

Les méthodes statistiques traditionnelles apprises sur de grandes quantités de données annotées et utilisées pour résoudre des tâches comme identifier le domaine, l'intention ou étiqueter les concepts en se basant uniquement sur les étiquettes des classes ne sont pas à même de pouvoir être utilisées

ici pour traiter de ces problèmes. Dans ce cas, d'autres informations doivent être exploitées pour extrapoler ces classes directement à partir de leurs définitions.

Dans le contexte d'une tâche comme le suivi de l'état du dialogue, l'adaptation multidomaine et l'adaptation à de nouveaux domaines non vus à l'apprentissage apportent des difficultés supplémentaires. Les résultats obtenus sur les corpus DSTC2, MultiWOz et SGD présentés en table 2 permettent d'apprécier l'évolution de la complexité de la tâche, selon que l'on utilise 1 domaine (DSTC2), plusieurs domaines (MultiWOz) ou plusieurs domaines dont des nouveaux domaines absents de la base d'entraînement (SGD).

	DSTC2	**MultiWOz**	**SGD**
Slot accuracy	97.5	96.9	96.5
Joint goal accuracy	74.5	48.6	25.4
Model	Zhong *et al.* (2018)	Wu *et al.* (2019)	Rastogi *et al.* (2019)

TABLE 2 – Tableau des résultats obtenus sur les corpus DSTC2, MultiWOz et SGD dans le contexte d'une tâche de suivi de l'état du dialogue mesurant la capacité des systèmes à réaliser la tâche (*joint goal accuracy* en anglais) ou à reconnaître les concepts (*slot accuracy* en anglais).

C'est sur la base d'états, construits par le gestionnaire de dialogue qu'une action est choisie pour être réalisée et c'est sur elles deux (états et actions) que se fonde l'optimisation automatique d'une politique d'interaction. Les actions représentent l'unité de communication fondamentale d'un système de dialogue qui définit les types d'interactions dont il est capable (demander la valeur d'un concept, demander la confirmation de celle-ci, etc.). À chaque tour, de nombreuses actions sont à la disposition du gestionnaire de dialogue. Dans SGD, les catégories d'actions possibles du système sont spécifiées à l'aide des 10 actes de dialogue (repris dans la table 3) et peuvent se combiner pour former des actions plus complexes.

Catégorie d'actions	**Description de l'action**	**Exemple de réponse du système**
CONFIRM	Confirme la valeur d'un slot avant d'effectuer un appel à un service transactionnel.	*Please confirm the update : A reservation at 10 :45 am in San Francisco.*
INFORM	Informe l'utilisateur.	*No, that ticket is not refundable.*
INFORM_COUNT	Indique le nombre de résultats trouvés qui répondent à la demande de l'utilisateur.	*There is 1 such flight. It is through Southwest Airlines.*
NOTIFY_SUCESS	Informe l'utilisateur que sa demande a abouti.	*Your reservation is made, and no they don't have any vegetarian options unfortunately.*
NOTIFY_FAILURE	Informe l'utilisateur que sa demande a échoué.	*Unfortunately I have been unable to make a reservation.*
OFFER	Renseigne l'utilisateur sur une certaine valeur pour un concept.	*Will Delta Airlines meet your requirement ?*
OFFER_INTENT	Propose à l'utilisateur une nouvelle intention.	*Would you like to purchase flight tickets from this airline ?*
REQUEST	Demande à l'utilisateur la valeur d'un concept.	*Any preferred location you would like to visit ?*
REQ_MORE	Demande à l'utilisateur s'il a besoin de quelque chose d'autre.	*Is there anything else you need help with ?*
GOODBYE	Met fin au dialogue.	*Have a nice day !*

TABLE 3 – Catégories d'actions, descriptions en langue naturelle et exemples de réponses du système.

Globalement, les tâches se caractérisent pas des niveaux de complexité variables. En ce qui concerne le suivi de l'état du dialogue conditionné par un schéma et la variabilité supplémentaire introduite dans le cas de schémas non vus dans la base d'entraînement, éloignés des schémas connus, associés à peu de données, on peut penser que faute d'avoir déjà rencontré les objectifs de la tâche, ou d'avoir appris à différencier suffisamment d'objectifs, le système pourra difficilement mettre en oeuvre des stratégies appropriées pour contrôler le déroulement de l'échange. Une stratégie se révèle appropriée quand elle permet d'atteindre un certain but et faire le choix d'une stratégie présuppose la connaissance du résultat visé et de la mesure de l'effort nécessaire pour y parvenir.

4 Conclusion et perspectives

Ces dernières années, l'apprentissage profond a permis de réaliser de nombreuses avancées dans des domaines variés où les données sont abondantes. Les modèles résultants sont dans une large mesure, spécialisés pour la tâche pour laquelle ils sont entraînés. Cette dépendance aux données se trouve être un obstacle majeur pour la portabilité vers de nouveaux domaines. Le développement de nouvelles méthodes fondées sur les données doit être adapté pour répondre à cette exigence. Par conséquent, la réduction de la quantité de données et d'annotations nécessaires pour l'entraînement des systèmes constitue une direction de recherche prioritaire dans le domaine des systèmes de dialogue.

Sans reprendre les enjeux portant sur l'éventail des capacités attendues pour un système, nous avons montré que ces capacités sont garantes d'échanges plus pertinents et de dialogues plus naturels. Nous avons présenté un tour d'horizon, succinct, de quelques approches de la littérature qui tentent de trouver un équilibre entre le connu et la nouveauté dans le processus d'acquisition de connaissances. Elles apportent des pistes intéressantes pour résoudre les problématiques auxquelles nous sommes confrontés en permettant de faciliter la conception des systèmes.

Enfin, le méta-apprentissage se trouve être un domaine actif d'intérêt croissant et nous aspirons à adapter ses algorithmes à la gestion du dialogue. Nous souhaitons pour cela étendre les réseaux de codes hybrides (Williams *et al.*, 2017) qui ont comme particularité de pouvoir concilier un apprentissage de bout en bout et l'intégration de codes métier (connaissances expertes) qui permettent de limiter certaines suites d'actions tout en réduisant considérablement la complexité d'apprentissage et la quantité de données requise pour l'entraînement. Pour ce faire il s'agira de repenser la représentation des actions et des états en considérant un espace «de plus haut niveau» tout en gardant la possibilité d'injecter de la supervision à l'aide de métarègles compatibles avec le paradigme du méta-apprentissage.

Références

BAPNA A., TÜR G., HAKKANI-TÜR D. & HECK L. (2017). Towards zero-shot frame semantic parsing for domain scaling. In *Proceedings of the 2017 INTERSPEECH Conference*, p. 2476–2480. DOI : 10.21437/Interspeech.2017-518.

BUDZIANOWSKI P., WEN T.-H., TSENG B.-H., CASANUEVA I., ULTES S., RAMADAN O. & GAŠIĆ M. (2018). MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 EMNLP Conference*, p. 5016–5026. DOI : 10.18653/v1/D18-1547.

CARUANA R. (1993). Multitask learning : A knowledge-based source of inductive bias. In *Proceedings of the 1993 ICML Conference*, p. 41–48. DOI : 10.1016/b978-1-55860-307-3.50012-5.

CHEN Y., HAKKANI-TÜR D. & HE X. (2016). Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *Proceedings of the 2016 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 6045–6049.

FERREIRA E., JABAIAN B. & LEFÈVRE F. (2015). Online adaptative zero-shot learning spoken language understanding using word-embedding. In *Proceedings of the 2015 ICASSP Conference*, p. 5321–5325. HAL : hal-02042298.

FINN C. (2018). *Learning to Learn with Gradients*. Thèse de doctorat, EECS Department, University of California, Berkeley.

HAKKANI-TÜR D., TÜR G., ÇELIKYILMAZ A., CHEN Y., GAO J., DENG L. & WANG Y. (2016). Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In N. MORGAN, Éd., *Proceedings of the 2016 INTERSPEECH Conference*, p. 715–719. DOI : 10.21437/Interspeech.2016-402.

HENDERSON M., THOMSON B. & WILLIAMS J. D. (2014). The second dialog state tracking challenge. In *Proceedings of the 2014 SIGDIAL Conference*, p. 263–272. DOI : 10.3115/v1/W14-4337.

KELLEY J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office information systems*, **2**(1), 26–41. DOI : 10.1145/357417.357420.

KIM S., GALLEY M., GUNASEKARA R. C., LEE S., ATKINSON A., PENG B., SCHULZ H., GAO J., LI J., ADADA M., HUANG M., LASTRAS L., KUMMERFELD J. K., LASECKI W. S., HORI C., CHERIAN A., MARKS T. K., RASTOGI A., ZANG X., SUNKARA S. & GUPTA R. (2019). The eighth dialog system technology challenge. *CoRR*, **abs/1911.06394**.

KUMAR A., MUDDIREDDY P., DREYER M. & HOFFMEISTER B. (2017). Zero-shot learning across heterogeneous overlapping domains. In *Proceedings of the 2017 INTERSPEECH Conference*, p. 2914–2918. DOI : 10.21437/Interspeech.2017-516.

LAROCHELLE H., ERHAN D. & BENGIO Y. (2008). Zero-data learning of new tasks. In *Proceedings of the 2008 AAAI Conference*, p. 646–651.

LEE S. & JHA R. (2019). Zero-shot adaptive transfer for conversational language understanding. In *Proceedings of the 2019 AAAI Conference*, p. 6642–6649. DOI : 10.1609/aaai.v33i01.33016642.

LIN T.-E. & XU H. (2019). Deep unknown intent detection with margin loss. In *Proceedings of the 2019 ACL Conference*, p. 5491–5496. DOI : 10.18653/v1/P19-1548.

LIU B. & LANE I. (2016). Attention-based recurrent neural network models for joint intent detection and slot filling. In *Proceedings of the 2016 INTERSPEECH Conference*, p. 685–689. DOI : 10.21437/Interspeech.2016-1352.

MI F., HUANG M., ZHANG J. & FALTINGS B. (2019). Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In S. KRAUS, Éd., *Proceedings of the 2019 IJCAI Conference*, p. 3151–3157. DOI : 10.24963/ijcai.2019/437.

PALATUCCI M., POMERLEAU D., HINTON G. E. & MITCHELL T. M. (2009). Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems*, p. 1410–1418.

PARISI G. I., KEMKER R., PART J. L., KANAN C. & WERMTER S. (2019). Continual lifelong learning with neural networks : A review. *Neural Networks*, **113**, 54 – 71. DOI : https ://doi.org/10.1016/j.neunet.2019.01.012.

RASTOGI A., ZANG X., SUNKARA S. K., GUPTA R. & KHAITAN P. (2019). Towards scalable multi-domain conversational agents : The schema-guided dialogue dataset. *To appear at AAAI 2020*.

SHALYMINOV I., SORDONI A., ATKINSON A. & SCHULZ H. (2020). Hybrid generative-retrieval transformers for dialogue domain adaptation. *CoRR*, **abs/2003.01680**.

THRUN S. & PRATT L., Éds. (1998). *Learning to Learn*. USA : Kluwer Academic Publishers.

TUR G., HAKKANI-TÜR D. & HECK L. (2014). Zero-shot learning and clustering for semantic utterance classification. In *Proceedings of the 2014 ICLR Conference*.

VANSCHOREN J. (2018). Meta-learning : A survey. *CoRR*, **abs/1810.03548**.

WANG W., ZHENG V. W., YU H. & MIAO C. (2019). A survey of zero-shot learning : Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, **10**(2). DOI : 10.1145/3293318.

WEN T.-H., GAŠIĆ M., MRKŠIĆ N., ROJAS-BARAHONA L. M., SU P.-H., VANDYKE D. & YOUNG S. (2016). Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the NAACL 2016 Conference*, p. 120–129. DOI : 10.18653/v1/N16-1015.

WILLIAMS J. D., ASADI K. & ZWEIG G. (2017). Hybrid code networks : practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 2017 ACL Conference*, p. 665–677. DOI : 10.18653/v1/P17-1062.

WU C.-S., MADOTTO A., HOSSEINI-ASL E., XIONG C., SOCHER R. & FUNG P. (2019). Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 808–819, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1078.

XIA C., ZHANG C., YAN X., CHANG Y. & YU P. (2018). Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 EMNLP Conference*, p. 3090–3099. DOI : 10.18653/v1/D18-1348.

XU P. & SARIKAYA R. (2013). Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, p. 78–83. DOI : 10.1109/ASRU.2013.6707709.

ZHANG X. & WANG H. (2016). A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the 2016 IJCAI Conference*, p. 2993–2999.

ZHONG V., XIONG C. & SOCHER R. (2018). Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1458–1467, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1135.

ZOPH B., YURET D., MAY J. & KNIGHT K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 EMNLP Conference*, p. 1568–1575. DOI : 10.18653/v1/D16-1163.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Simplification de textes : un état de l'art

Sofiane ELGUENDOUZE[1,2]
(1) LS2N, Nantes, France
(2) ESI, Alger, Algérie
fs_elguendouze@esi.dz

RÉSUMÉ

Cet article présente l'état de l'art en simplification de textes et ses deux grandes familles d'approches, à savoir les approches à base de règles et les approches statistiques. Nous présentons, en particulier, les récentes approches neuronales et les architectures mises en place ainsi que les méthodes d'évaluation des systèmes de simplification.

ABSTRACT

Text simplification (State of the art)

This paper presents the state of the art on text simplification, in particular the two main types of approaches, namely rule-based approaches and statistical approaches (or data-driven). We present, in particular, the recent neural approaches and the architectures implemented as well as the evaluation methods of text simplification systems.

MOTS-CLÉS : Simplification de textes, Apprentissage automatique, Apprentissage profond, Traduction automatique, Lexique, Syntaxe, Discours.

KEYWORDS: Text simplification, Machine learning, Deep learning, Machine translation, Lexicon, Syntax, Discourse.

1 Introduction

Les textes incorporent des constructions linguistiques complexes, ce qui peut entraîner des difficultés de lecture et/ou de compréhension chez les personnes avec des compétences linguistiques réduites comme les apprenants d'une langue non native ou les personnes ayant un trouble du langage telles que les autistes, les aphasiques, les dyslexiques etc.
La simplification de Texte (ST) est définie comme un processus permettant de détecter les phénomènes problématiques dans un texte, causant des difficultés de lecture et/ou de compréhension, et de procéder à l'adaptation de son contenu pour les résoudre. Cela peut être effectué par le remplacement des mots difficiles par des synonymes plus faciles à comprendre, le changement des temps de conjugaison difficiles par d'autres plus courants ou encore par la résolution d'anaphores. Tout cela a pour objectif de rendre les textes plus accessibles compréhensibles par une catégorie de lecteurs spécifiques.

Les premiers travaux de recherche sur la simplification de textes ont commencé dans les années 90, avec l'approche de simplification syntaxique proposée par (Chandrasekar *et al.*, 1996), qui visait à améliorer les performances des analyseurs syntaxiques en langage naturel. Des recherches ultérieures ont principalement porté sur la ST pour des catégories d'utilisateurs avec des déficiences intellectuelles spécifiques. Certains travaux se sont concentrés sur l'aphasie (Carroll *et al.*, 1998),

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 96–109.
HAL : hal-02786191.

la dyslexie (Quiniou & Daille, 2018), l'autisme (Štajner *et al.*, 2012), les personnes avec un faible niveau de lecture (Williams & Reiter, 2008), les personnes sourdes (Inui *et al.*, 2003), les apprenants d'une langue étrangère (Siddharthan & Katsos, 2010). L'intérêt porté à la simplification de texte a récemment augmenté grâce à la disponibilité des textes et l'explosion du web etc., et le nombre de langues sur lesquelles elle est appliquée ne cesse d'accroître : l'anglais (Siddharthan, 2006; Zhu *et al.*, 2010; Xu *et al.*, 2016), le français (Seretan, 2012; Brouwers *et al.*, 2012, 2014; Gala *et al.*, 2018), l'espagnol (Saggion *et al.*, 2015), le portugais (Aluísio & Gasperin, 2010) ...
L'objectif de ce papier est de mettre en avant les dernières techniques de simplification notamment les approches neuronales qui ont largement contribué à l'amélioration des systèmes de simplification. La suite du papier présentera une typologie des phénomènes problématiques nécessitant une simplification, les différents types approches et les mesures d'évaluation les plus utilisées actuellement.

2 Taxonomie des problèmes

La complexité des textes et les informations implicites contenues dans ceux-ci affectent une tranche considérable de la population qui souffre de difficultés de lecture et de compréhension. Cela montre qu'il existe une multitude de domaines pour lesquels la simplification de textes pourra être utilisée. Plusieurs phénomènes problématiques tels que les phrases longues, les phrases complexes ou encore les pronoms et leurs référents implicites, sont généralement difficiles à comprendre par le lecteur. Des constructions syntaxiques telles que les phrases qui ne respectent pas la forme canonique (sujet, verbe, complément) peuvent également être problématiques pour les personnes aphasiques ou autistes. Il en va de même pour le vocabulaire très difficile ou spécialisé et les mots rares qui peuvent être ambigus. Les apprenants d'une langue étrangère peuvent avoir un lexique très restreint et ne seraient donc pas en mesure de comprendre certaines constructions grammaticales complexes ou certains mots difficiles. La Table 1 présente une synthèse d'analyse de certaines typologies de phénomènes linguistiques précédemment proposées par (Brouwers *et al.*, 2012; Gala & Ziegler, 2016).

3 Simplification de textes

De la méthode manuelle classique aux méthodes automatiques, la simplification a connu une forte croissance, en termes de pertinence des résultats obtenus et de champs d'applications pour lesquels de tels systèmes ont été conçus.

3.1 Simplification manuelle

La simplification manuelle de textes adopte deux grandes approches pour simplifier un texte, la première consiste à créer un texte en suivant des recommandations linguistiques et textuelles destinées à faciliter la compréhension en fonction du type de public visé (la collection « La traversée », par exemple, s'adresse à des adultes débutants en lecture ou faibles lecteurs). La deuxième approche consiste à transformer un texte original, jugé difficile à comprendre, en un texte plus simple, destiné à un public particulier. L'exemple le plus connu est Vikidia, un ensemble de textes encyclopédiques destinés aux enfants et directement inspiré des articles de Wikipédia. Plusieurs études ont souligné

Niveau	Phénomène	Explication
Lexical	Termes issus d'une langue étrangère	Termes anglais dans un texte français
	Termes difficiles	Termes hors vocabulaire
	Termes non pertinents à la compréhension	Adjectifs, adverbes etc.
Syntaxique	Temps de conjugaison moins courants et plus littéraires	Imparfait, plus que parfait, passé simple etc.
	Éléments grammaticaux secondaires ou redondants	Propositions subordonnées, adverbiales, relatives, compléments circonstanciels
	Structures syntaxiques complexes	Forme négative, discours indirect, forme passive (forme non canonique)
Discursif	Organisation compliquée de l'information	Pour des raisons esthétiques ou autres
	Cohérence et cohésion	Anaphores (nominales et pronominales), anaphores difficiles (chaînes anaphoriques, dialogues etc.)
	Informations secondaires	Trop d'exemples, de définitions etc.
	Manque d'informations	Manque d'exemples, de définitions etc.

TABLE 1 – Typologie des phénomènes problématiques en ST

le rôle de la simplification manuelle de textes pour la compréhension tel que(Anderson & Davison, 1986) etc.

3.2 Simplification automatique

La simplification automatique de textes (SAT) comporte deux tâches principales : la simplification lexicale (SL) et la simplification syntaxique (SS). Celles-ci peuvent être traitées séparément ou conjointement. La SL modifie le vocabulaire du texte en substituant par exemple les termes jugés difficiles par des synonymes ou paraphrases qui sont plus simples à comprendre. Par exemple, la phrase « *parce que sa femme le **gouvernait** entièrement* » pourra être remplacée par « *parce que sa femme le **dirigeait** entièrement* ». D'autre part, la SS a pour objectif de convertir les phrases contenant des structures syntaxiques complexes tels que les propositions subordonnées et les phrases en forme passive en phrases plus simples structurellement en préservant leur sens original (ou au moins en limitant son altération). Par exemple, « *Il reconnut que le cinquième compartiment avait été envahi par la mer* » ; pourra être remplacée par « *Il reconnut que la mer a envahi le cinquième compartiment* ». La SAT comporte trois principales taches : (i) la détection automatique des phénomènes problématiques et des éléments linguistiques complexes impliquant des aspects lexicaux, syntaxiques ou autres (cf. Table 1) (ii) la production d'une version simplifiée des textes à l'aide d'un ensemble d'opérations de simplification (iii) l'évaluation des simplifications apportées.

Dans le domaine de SAT, la majorité des travaux se limite aux traitement de phénomènes problématiques lexicaux ou syntaxiques. Très peu d'attention a été portée au niveau discursif.

4 Approches de simplification

Les approches de SAT peuvent être regroupées en trois grandes familles, notamment les approches par règles classiques, les approches statistiques et finalement la combinaison des deux donnant naissance aux approches hybrides.

4.1 Approches par règles

Ce sont les toutes premières approches adoptées en SAT, elles ont été proposées pour des cas d'application spécifiques et pour une population bien ciblée. (Inui *et al.*, 2003) ont réalisé un système de simplification pour les personnes sourdes, en considérant les deux niveaux lexical et syntaxique. (Williams & Reiter, 2008) ont proposé un système de génération de textes simplifiés au niveau discursif appelé SKILL-SUM pour les personnes avec faible niveau d'alphabétisation, en favorisant les mots et les phrases courtes qui sont plus lisibles et plus compréhensibles par les lecteurs .

La simplification s'effectue sur la base de règles définies explicitement par des experts, en analysant des textes originaux et leurs équivalents simplifiés. Par exemple, (Brouwers *et al.*, 2014) ont utilisé 19 règles pour effectuer une simplification syntaxique sur des textes français, classés en 3 catégories : Suppression (12 règles), Modification (3 règles) et Division (4 règles). La suppression par exemple procède à l'élimination directe des éléments syntaxiques secondaires comme les propositions subordonnées. Exemple : la phrase « *Le candidat a proposé une idée intéressante, qui parait très originale.* » devient « *Le candidat a proposé une idée intéressante* ».

Bien qu'elle soit la plus ancienne, cette approche reste d'actualité notamment pour les langues peu dotées pour lesquelles il n'existe pas de corpus parallèles. Elle sont très adaptées à la simplification syntaxique. Leur principal inconvénient est toutefois une portabilité et une évolutivité réduites pour les nouveaux scénarios, qui nécessitent la création de nouveaux ensembles de règles à chaque fois qu'une nouvelle langue (ou un nouveau domaine) doit être couvert. De plus :

1. Elles consomment beaucoup de temps.
2. Elles requièrent beaucoup d'implication humaine pour la définition des règles.
3. Il est impossible de trouver et énumérer toutes les règles de simplification.

4.2 Approches statistiques

Ces approches reposent principalement sur la disponibilité de grands corpus utilisés à la place de connaissances expertes. L'objectif est d'apprendre les règles automatiquement depuis ces ressources de données.

L'approche de simplification par **transduction d'arbres** (Tree Transduction en anglais) vise à sur-générer des règles de simplification automatiquement, ensuite à choisir celles qui correspondent le mieux. (Paetzold & Specia, 2013) proposent une approche composée de trois modules : Un module d'entraînement qui sur-génère des règles de transformation candidates, pour cela il reçoit en entrée les représentations sous forme d'arbres syntaxiques du corpus parallèle aligné au niveau des mots, effectue les différentes transformations possibles sur l'arbre représentant la phrase complexe pour reproduire l'arbre représentant la phrase simplifiée, ce qui nous donne en sortie les différentes règles candidates (lexicales, syntaxiques, lexico-syntaxiques). Le module sélectionne ensuite les

transformations les plus susceptibles de représenter des opérations de simplification en vérifiant certains critères. Le module de simplification génère à partir d'une phrase complexe en entrée et des règles précédemment sélectionnées, des phrase simplifiées candidates. Finalement, le module de classement attribue des scores aux différentes candidates et les ordonne pour sélectionner la meilleure.

Les approches par **traduction automatique (TA)** (**MT** pour Machine Translation) considèrent la simplification de texte comme un problème de traduction monolingue. Initialement, elles ont été proposées pour faire de la traduction d'une langue en une autre, plus tard elles ont été utilisées pour faire la simplification de texte (Specia, 2010). Le début des années 90 a vu le lancement des approches de traduction automatique statistique (SMT pour Statistical machine translation) où les règles ainsi que les modèles de simplification peuvent être appris automatiquement à partir de corpus parallèles constitués de paires de phrases (Complexes-Simplifiées). Elles regroupent les méthodes suivantes :

— **Traduction automatique statistique fondée sur les mots** (Word Based Statistical Machine Translation) est la plus ancienne de ces méthodes et la moins utilisée, l'unité fondamentale étant le mot, les résultats obtenus étaient moins corrects vu qu'elle effectue une traduction mot à mot en ignorant l'aspect syntaxique et l'aspect sémantique bien entendu.
— **Traduction automatique statistique fondée sur les syntagmes** (Phrase Based Statistical Machine Translation) Le but étant de manipuler des séquences de mots (de taille variable) lors de la traduction à la place de mots seuls. La méthode consiste à segmenter une phrases en syntagmes, ensuite à faire la traduction syntagme par syntagme puis les réordonner pour formuler une phrase de sortie a priori simplifiée. Certains travaux avaient pour objectif la fragmentation des phrases longues en phrases plus courtes et plus simples (Specia, 2010), d'autres se sont intéressés à la suppression des expressions secondaires (Coster & Kauchak, 2011).
Cette méthode ne peut effectuer qu'un petit nombre d'opérations de simplification, telles que la substitution lexicale, la suppression et l'explication simple. Elle n'est pas bien adaptée aux opérations de réorganisation ou de fragmentation.
— **Traduction automatique statistique fondée sur la syntaxe** (Syntax Based Statistical Machine Translation) La différence avec la méthode précédente est qu'elle manipule des unités syntaxiques complètes au lieu de mots ou de syntagmes seuls, incorporant ainsi une représentation explicite de la syntaxe dans les systèmes MT (comme l'ordre des mots par exemple), et permettant donc d'effectuer de meilleures opérations de réorganisations. Exemples de travaux : (Zhu *et al.*, 2010; Xu *et al.*, 2016).

L'inconvénient principal des méthodes de traduction automatique statistiques est qu'elles fonctionnent séparément sur de petits composants de simplification (lexical seulement ou syntaxique seulement). De plus, nous trouvons que parfois, elles traitent partiellement un niveau linguistique, comme la fragmentation uniquement par rapport au niveau syntaxique.

La **Traduction automatique neuronale** (NMT pour Neural Machine Translation) est récemment apparue (Kalchbrenner & Blunsom, 2013) en tant qu'une nouvelle approche de TA, et a montré une forte amélioration par rapport aux approches précédentes classiques. De plus, contrairement aux méthodes traditionnelles qui fonctionnent séparément sur de petits composants, NMT réalise une simplification de bout en bout (end-to-end), cela veut dire qu'elle ne nécessite pas des décodeurs

externes, des modèles de langage [1] etc.
L'approche centrale de NMT est l'architecture d'encodeur-décodeur implémentée par les réseaux de neurones récurrents (RNN), la séquence d'entrée est ainsi représentée par un vecteur (encodage), puis décodée pour obtenir la séquence de sortie représentant une phrase simplifiée. Par exemple, (Zhang & Lapata, 2017) ont introduit une architecture encodeur-décodeur à base de RNN, couplée à un modèle d'apprentissage par renforcement. L'encodeur-décodeur est considéré comme Agent du modèle d'apprentissage par renforcement. Etant donné une phrase complexe en entrée, l'encodeur la transforme en une séquence d'états cachés avec un réseau de neurones LSTM (Long Short Term Memory), et à chaque étape t il réalise une action $\hat{y}t$ appartenant à un vocabulaire fixe de sortie, suivant une certaine politique. L'agent continue à faire des actions, jusqu'à ce qu'il produise la fin de la phrase. La sortie simplifiée de l'encodeur-décodeur serait ainsi $\hat{Y} = (\hat{y}1, \hat{y}2, \hat{y}3 \ldots)$. Une récompense r est alors attribuée, et l'algorithme de renforcement met à jour l'agent. Y est la sortie 'Référence', elle est utilisée pour l'évaluation de la simplicité de la sortie simplifiée générée et le calcul de la récompense r.

Un problème avec toutes les approches statistiques, est que la pertinence des résultats dépend largement de la taille des corpus parallèles utilisés, dont leur construction est en général très coûteuse. Certaines solutions ont été proposées pour contourner ce problème de manque de ressources notamment le travail de (Aprosio *et al.*, 2019) où ils ont construit un nouveau modèle encodeur-décodeur basé sur l'attention (Vaswani *et al.*, 2017). Leur idée repose sur l'extension de petits corpus d'entraînement avec des données synthétiques pour satisfaire le besoin en données volumineuses nécessaires à l'entraînement de leur modèle. Le modèle de simplification fournit une solution de bout en bout pour traiter à la fois la simplification lexicale et syntaxique, ceci en apprenant à faire des changements structurels plus complexes que les changements terme à terme uniquement.

(Aprosio *et al.*, 2019) ont suivi trois stratégies différentes pour l'augmentation (extension) des données, à partir des données « de référence » (gold) à leur disposition et qui sont en quantité limitée. La première stratégie est le sur-échantillonnage, celle-là consiste à faire augmenter la taille du corpus d'entraînement en dupliquant le corpus original plusieurs fois afin de maximiser l'exploitation des paires de phrases références à leur disposition. La deuxième consiste à créer des paires synthétiques simple-simple à partir de grands corpus monolingues, ceci est réalisé en extrayant automatiquement les phrases les plus simples avec des méthodes heuristiques, ensuite les dupliquer pour créer des paires simple-simple, celles-ci sont ensuite utilisées comme données synthétiques pour former le système de simplification. L'objectif étant d'introduire un biais dans le décodeur en vue d'améliorer la simplicité des sorties. La troisième stratégie est la création de paires synthétiques complexe-simple pour former un système complexifiant qui génère des phrases complexes à partie de celles simples précédemment sélectionnées (en utilisant des outils d'Open-NMT). Les paires sont ensuite inversées pour former le système de simplification. L'intuition étant de conserver les phrases simplifiées générées par l'expertise humaine dans la partie cible des données parallèles afin d'améliorer la génération de phrases simplifiées.

Leur système de simplification fonctionne de la manière suivante : Initialement, une séquence de mots est transmise à l'encodeur. A chaque pas de temps, sur la base des représentations générées par l'encodeur et du mot généré dans le pas de temps précédent, le décodeur génère le mot suivant. Ce

1. Un modèle de langage correspond à une fonction, qui prend une phrase traduite et renvoie la probabilité qu'elle soit dite par un locuteur natif, de plus elle peut aider à choisir une traduction parmi plusieurs pour un mot donné, selon son contexte.

processus se poursuit jusqu'à ce que le décodeur génère le symbole de fin de phrase. Un module de génération de pointeurs est rajouté à ce modèle, il permet à la fois de copier des mots à partir de la phrase source et de générer des mots à partir d'un vocabulaire fixe partagé contenant tous les mots des phrases d'apprentissage complexes et simples. À chaque pas de temps, le réseau estime la probabilité de générer un mot et utilise cette probabilité pour décider de générer ou de copier le mot.

Récemment, des techniques impliquant de l'apprentissage non supervisé ont vu le jour. (Surya *et al.*, 2018) ont conçu un modèle avec un encodeur partagé et deux décodeurs à base d'attention, en plus d'un discriminateur pour influencer le comportement du décodeur et d'un classifieur pour la diversification, selon la nature de l'entrée (simple/complexe). Le modèle reconstitue d'un coté l'entrée originale complexe à partir de la phrase simplifiée, et de l'autre génère une version simplifiée de l'entrée. Ceci est réalisé en examinant la structure et les schémas linguistiques d'un grand nombre de phrases simples et complexes non alignées (à partir de Wikipédia) qui sont beaucoup moins coûteuses et faciles à obtenir que les données parallèles alignées.

4.3 Approches hybrides

L'objectif de ces approches est de tirer profit des avantages des deux approches précédentes (par règles et statistiques) en combinant la simplification syntaxique par règles, et la simplification lexicale par approches statistiques. Ceci en fait est dû aux limitations de la simplification lexicale basée sur les règles (trop de règles à énumérer manuellement), et celles de la simplification syntaxique statistique (qui produit des résultats moins bons grammaticalement et qui sont très limitées dans leur portée, comme le cas du passage de la voix passive à la voix active, chose qui est beaucoup plus pertinente par règles). Un exemple de travaux est celui de (Siddharthan & Mandya, 2014), qui ont proposé un système hybride combinant un module de simplification lexicale statistique, et un module de simplification syntaxique basée sur 136 règles.

Les approches neuronales sont considérés plus performantes, en effet elles traitent la simplification de textes de bout en bout, d'une façon plus rapide et en donnant des résultats plus pertinents contrairement aux méthodes hybrides modulaires où les résultats sont relativement moins pertinents pour certaines opérations.

5 Evaluation des systèmes de simplification de textes

L'évaluation des systèmes de SAT vise généralement à tester leur pertinence en termes de qualité des sorties, ou à évaluer leur efficacité/utilité en mesurant le temps de lecture et la compréhension des lecteurs (utilisateurs finaux). Cependant, ce n'est pas toujours possible de procéder à un test réel effectué par la population cible comme les dyslexiques où les apprenants de langue. Une évaluation par des experts du domaine est alors effectuée, en attribuant des scores sur la simplicité des résultats, leur pertinence en termes grammaticalité, et leur préservation du sens. Cette méthode est dite manuelle ou humaine, et elle n'opère qu'au niveau de la phrase. Des méthodes automatiques sont alors apparues pour faciliter la tâche aux concepteurs de systèmes de simplification, en leur économisant du temps et de coût. Ces évaluations automatiques s'appliquent sur le texte entier, en s'appuyant par exemple sur des mesures dédiées aux systèmes MT ou des formules spécifiques. La figure 1 montre un schéma synthétisant les techniques l'évaluation existantes.
Il est à noter que l'évaluation s'effectue sur les textes simplifiés aussi bien que sur les textes originaux

complexes. Pour le premier cas comme nous l'avons mentionné plus haut, elle mesure la lisibilité des textes pour tester l'efficacité du système de simplification et la qualité de ces sorties. La deuxième vise à identifier à partir du texte original, les parties qui sont particulièrement complexes et qui doivent par conséquent être simplifiées.
La qualité du résultat généré par les systèmes de simplification de textes est généralement évaluée en utilisant une combinaison de mesures de lisibilité automatiques (mesure du degré de simplicité principalement) et d'évaluation humaine (mesure de grammaticalité, de simplicité et de préservation du sens).

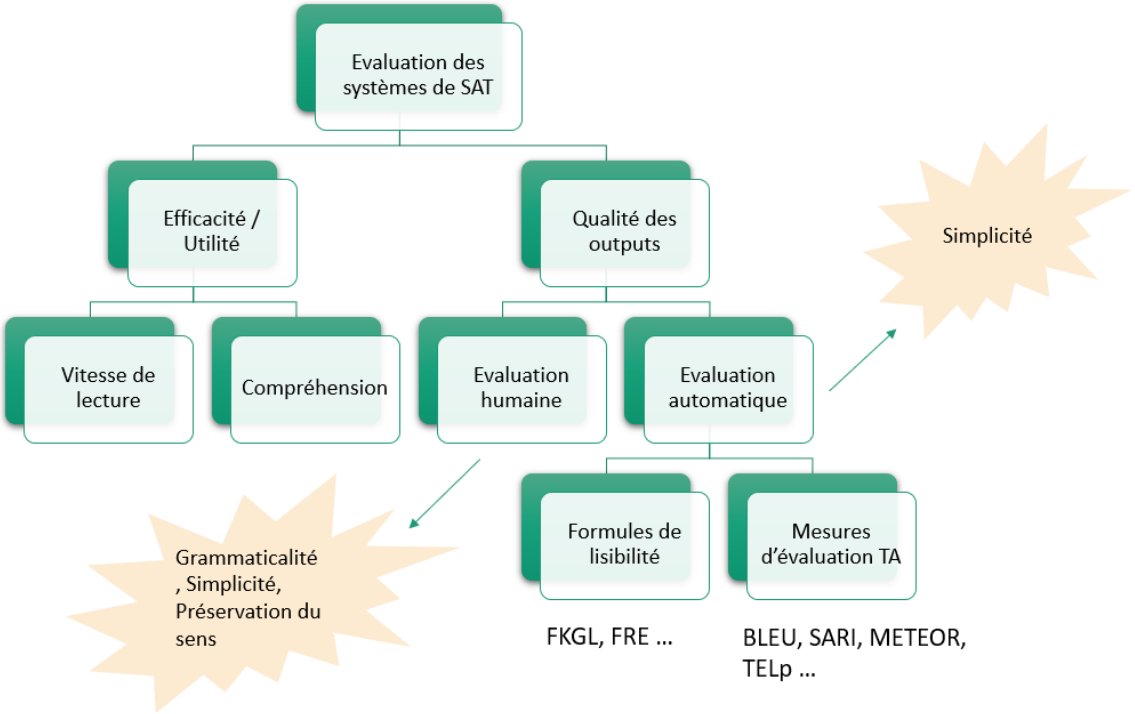


FIGURE 1 – Évaluation des systèmes de SAT. Schéma inspiré de (Štajner & Saggion, 2018)

5.1 Évaluation manuelle

L'évaluation manuelle (EM) se fait en mesurant certains critères sur les textes simplifiés en sortie à savoir : l'aisance (ou la facilité) de lecture qui mesure l'exactitude et la grammaticalité, le degré de simplicité et la préservation du sens qui mesure la correspondance du sens de la phrase simplifiée à celui de la phrase originale. En général, ces mesures sont données sur une échelle de 1 à 5. Plus le score est élevé, plus la qualité du texte simplifié en sortie est meilleure.

5.2 Évaluation automatique

Comme pour les nouvelles méthodes de SAT, leur évaluation en parallèle a eu sa part d'automatisation. En effet, l'EM présente quelques inconvénients, notamment le coût élevé, la consommation du temps, le manque d'adaptabilité, et la non-reproductibilité. L'évaluation automatique (EA) vient alors non pas pour remplacer l'EM mais plutôt pour la compléter. En effet, parfois elle ne mesure que le degré

de simplicité des sorties, contrairement à l'EM qui mesure la grammaticalité, la préservation du sens et la simplicité. Elle s'applique de l'une des manières suivantes :

Formules de lisibilité Depuis 1950, plus de 200 formules d'évaluation ont été développées pour l'anglais, or ce n'était pas le cas pour le Français où on ne retrouve que quelques travaux comme celui de (François & Watrin, 2011). Les toutes premières formules ont été calculées sur la base de la longueur des mots et la longueur moyenne des phrases seulement, parce que ces deux paramètres se corrèlent bien avec les tests de lecture, parmi ces formules : FKGL 'Flesch-Kincaid Grade Level index' (Kincaid *et al.*, 1975) ; FRE 'Flesch Reading Ease score' (Flesch & Gould, 1949) ; Ces formules sont restées largement utilisées jusqu'à maintenant. D'autres formules comme Dale-Chall (Dale & Chall, 1948) ont été calculées sur la base de la longueur moyenne des phrases et de la proportion de mots hors le vocabulaire simple.
Cependant, ces formules présentent plusieurs lacunes. Par exemple, elles ne prennent en compte que les caractéristiques superficielles, en ignorant d'autres aspects importants contribuant à la difficulté du texte, tels que la cohérence, la densité du contenu, la capacité d'inférence [2] du lecteur etc. Ils omettent également l'aspect interactif du processus de lecture.

L'avancement puissant en méthodes du TAL et des techniques du ML ont fait émerger de nouvelles approches pour l'évaluation de la lisibilité qui donnent de meilleurs résultats par rapport aux formules précédentes. (Petersen & Ostendorf, 2009) ont utilisé des modèles de langues [3] et des SVM avec de nouveaux attributs tels que la hauteur moyenne de l'arbre syntaxique, le nombre moyen de phrases nominales et verbales, nombre moyen de syllabes par mot etc. (Feng *et al.*, 2010), a montré que d'autres attributs tels que la longueur des chaînes lexicales, la densité des entités nommées dans le document, la longueur moyenne des phrases (qui est plus utile et moins coûteuse), se corrèlent mieux que le FKGL avec la compréhension du lecteur, notamment les apprenants d'une langue étrangère et les enfants avec troubles de lecture. (François & Fairon, 2012) ont présenté une nouvelle formule d'évaluation de la lisibilité pour 'le Français comme langue étrangère (FLE)', en utilisant les SVMs et 46 caractéristiques textuelles couvrant les trois niveaux lexical, syntaxique et sémantique, ainsi que des caractéristiques spécifiques au FLE tels que la fréquence moyenne des expressions multi-mots dans le texte, la nature du texte qui s'obtient en calculant certains indicateurs (le taux de ponctuation, présence des virgules etc.). (Vajjala & Meurers, 2014) ont utilisé un modèle de régression entraîné sur des documents entiers pour comparer la lisibilité des phrases parallèles alignés du corpus Wikipédia-SimpleWikipédia (Zhu *et al.*, 2010).

Mesures des systèmes de traduction automatique En raison du développement large des systèmes de simplification de TA, il est devenu de plus en plus indispensable d'évaluer leurs performances, non seulement pour établir une comparaison, mais aussi pour savoir s'ils réalisaient des progrès. Cependant, l'évaluation de la traduction automatique est difficile car les langues naturelles sont très ambiguës et le contenu s'exprime différemment d'une langue à une autre.
Les techniques actuelles se basent principalement sur la comparaison entre les résultats produits par les systèmes de simplification et les textes simplifiés manuellement considérés 'de référence', ici on distingue différentes manières de faire : soit en fournissant une seule référence ou bien en donnant des références multiples pour améliorer la précision de la comparaison. D'autres mesures n'utilisent

2. Une opération mentale qui permet au lecteur de déduire les non-dits ou les éléments implicites dans un texte
3. Prédisent la probabilité qu'une séquence de mots particulière se produise

pas de textes de référence. En général, les mesures des systèmes MT évaluent la **similitude lexicale** par le moyen de l'ordre des mots dans les phrases, la distance de modification, et le chevauchement des séquences des mots. Des **caractéristiques linguistique** sont également pris en compte, telles que la syntaxe et la sémantique (l'étiquetage morpho-syntaxique ou POS-tagging, les structures des phrases, les synonymes, les entités nommées, les rôles sémantiques et les modèles de langage etc.) Ces deux catégories sont généralement inséparables, en effet certaines mesures de la première catégorie utilisent certaines caractéristiques linguistiques et vice versa. Les dernières recherches appliquent des modèles d'**apprentissage profond** pour l'évaluation . Nous présentons en particulier certaines mesures d'évaluation pour la similarité lexicale :

1. **Distance des modifications**
   Elle se calcule en comptant le nombre minimum de changements à apporter au texte simplifié pour le transformer en texte de référence. On trouve :
   — WER (Word Error Rate) : introduit par (Su *et al.*, 1992), elle tient compte de l'ordre des mots. Les opérations possibles sont l'insertion, la suppression et la substitution des mots. Le nombre minimum des modifications est calculé par la formule suivante :

$$WER = \frac{N(Substitutions) + N(Insertions) + N(Suppressions)}{LongueurRef}$$

   tel que : N(x) est le nombre d'opérations x effectuées et LongueurRef est la longueur de la phrase de référence. Une des faiblesses de WER est qu'elle ne tient pas compte de l'ordre des mots correctement, en effet elle est très réduite lorsque l'ordre d'un mot ne correspond pas à celui du texte de référence (Fausses phrases).
   — PER (Position-independent Word Error Rate) : proposée par (Tillmann *et al.*, 1997) pour résoudre ce problème de fausses phrases, en ignorant l'ordre des mots lors de la reconstitution de la phrase de référence. La mesure se fait en calculant le nombre de fois qu'un mot apparaît identiquement dans les deux phrases (simplifié-référence) qui est donné par le paramètre 'Correct', et selon la différence en longueur entre les deux phrases, le reste des mots sont insérés ou supprimés. Voici la formule correspondante :

$$PER = 1 - \frac{Correct - Max(0\ ,\ LongSortie - LongRef)}{LongRef}$$

2. **Précision et rappel**
   Ce sont deux propriétés qui ont été confirmées par plusieurs mesures comme étant essentielles pour une corrélation élevée avec les jugements humains. Nous introduisons :
   — BLEU (Bilingual Evaluation Understudy) : est la plus utilisée et la moins coûteuse, proposée par (Papineni *et al.*, 2002). Elle montre une corrélation élevée avec les jugements humains pour la grammaticalité et la préservation du sens, elle n'est cependant pas bien adaptée pour l'évaluation de la simplicité des résultats. Le calcul du score se fait en premier lieu au niveau de chaque segment du texte simplifié (généralement une phrase), ensuite un score total correspondant à la moyenne géométrique est calculé, elle est basée sur le calcul de la précision pour des n-grammes de taille 1 à 4 avec un coefficient de pénalité de brièveté (BP), et donne des poids égaux pour les différents n-grammes. Le résultat est toujours compris entre 0 et 100 %, plus il est proche de 100, meilleure est la simplification. Cette mesure pénalise fortement la réorganisation des mots et le raccourcissement des phrases. Sa formule étant :

$$BLEU = BP * \exp \sum_{i=1}^{n} \lambda_n \log Precision_n$$

$$BP = \begin{cases} 1 & \text{si } c > r \\ \exp^{(1-r/c)} & \text{si } c \leq r \end{cases}$$

Où **"c"** est la longueur totale de la phrase simplifiée ; **"r"** la longueur de la phrase de référence, et si plusieurs références existent, celle de longueur la plus proche à celle de la phrase simplifiée est choisie ; λ_n sont des poids positifs de précision pour les n-grammes (leur somme est à 1) généralement pris identiques.

— SARI : proposée récemment par (Xu *et al.*, 2016). Elle mesure la simplicité à travers des opérations d'ajout, de suppression et de conservation. Elle se différencie par le fait que la phrase simplifiée est comparée aux phrases de référence ainsi que la phrase d'origine non simplifiée. SARI a montré une forte corrélation avec les jugements humain pour la simplicité et est actuellement la principale mesure utilisée pour évaluer les modèles de simplification. Sa formule :

$$SARI = c1 * F_{ajout} \ + \ c2 * F_{conservation} \ + \ c3 * F_{suppression}$$

$$F_{oper} = \frac{2*P_{oper}*R_{oper}}{R_{oper} + P_{oper}} \ \text{|} \ P_{oper} = \frac{1}{k} * \sum_{n=1}^{k} p_{oper(n)} \ \text{|} \ R_{oper} = \frac{1}{k} * \sum_{n=1}^{k} r_{oper(n)}$$

Tel que : c1 = c2 = c3 = 1/3 ; **"k"** représente l'ordre le plus grand des n-grammes ; $oper \in \{ajout, conservation, suppression\}$; $p_{oper(n)}$ et $r_{oper(n)}$ représentent respectivement la précision et le rappel des n-grammes correspondants à l'opération en question.

— F-mesure : c'est une combinaison de la précision P et du rappel R, elle était d'abord adoptée par la recherche d'information, ensuite par l'extraction d'information et l'évaluation des systèmes de simplification TA. La formule utilisée étant :

$$F_\beta = (1 \ + \ \beta^2) \ \frac{P*R}{R + \beta^2 * P}$$

3. **Ordre des mots**
L'ordre des mots est un facteur très significatif pour évaluer la similarité lexicale. La diversité linguistique permet toutefois différentes apparences ou structures pour une phrase, le défi est donc de pouvoir appliquer la pénalité sur les mots qui sont vraiment incorrects (des phrases mal structurées) plutôt que sur des mots corrects avec un ordre différent. Au contraire, une phrase simplifiée ayant un ordre de mots différent de celui de la référence mais correcte est un vrai point d'intérêt, vu que ça permettra éventuellement une meilleure représentation de l'information dans la phrase d'une manière pouvant être plus claire est plus simple que dans la phrase référence. On trouve les mesures : ATEC (Assessment of Text Essential Characteristic), PORT (Precision-Order-Recall MT Evaluation Metric for Tuning), LEPOR (Length Penalty, Precision, n-gram Position difference Penalty and Recall) etc.

6 Conclusion et perspectives

Nous avons présenté à travers cet article, l'état de l'art en simplification de texte. Cette dernière est définie comme un processus permettant de modifier des textes difficiles afin de les rendre plus simples et plus accessibles à certaines catégories de lecteurs. Nous avons élaboré en premier lieu une typologie de phénomènes problématiques comprenant trois niveaux linguistiques (Lexical, syntaxique

et discursif). Ensuite nous avons abordé les différents types d'approches pour la simplification, notamment l'approche par règles, l'approche statistique et l'approche hybride. Les approches statistiques consistent actuellement à faire en grande partie de l'apprentissage automatique/profond qui prennent de plus en plus de l'ampleur en TAL, notamment en simplification de textes. Finalement, nous avons introduit certaines mesures d'évaluation pour les systèmes de simplification telles que les formules de lisibilité et les mesures des systèmes MT.
La simplification de textes implique toutefois des problèmes majeurs liés à l'apprentissage et à la compréhension, vu qu'elle introduit des erreurs et des ambiguïtés lors de la modification du texte original. De plus, une simplification excessive risque de limiter l'apprentissage et la transmission d'informations. Nous travaillons par conséquent sur un nouveau paradigme que nous appelons l'explicitation de textes, et qui permet de ne pas modifier le texte original. Cela consiste en revanche à adapter les approches de simplification de bout en bout (qui intègrent à la fois la détection et la simplification des difficultés), de sorte à faire la détection mais à traiter les phénomènes problématiques différemment, en enrichissant le texte original d'éléments explicitant ses informations implicites et difficiles. Les architectures récentes à base de transformeurs peuvent être très intéressantes dans le cadre de l'explicitation, et notre travail consiste à proposer un tel système d'explicitation pour deux catégories de lecteurs (Les enfants dyslexiques et les apprenants du français comme langue étrangère).

Remerciements

Nous remercions Solen QUINIOU et Béatrice DAILLE pour leur encadrement et leur appui scientifique. Nous remercions Lynda SAID LHADJ pour son support et pour le partage de savoir-faire et de connaissances. Merci à tous les trois pour la relecture de l'article.

Références

ALUÍSIO S. M. & GASPERIN C. (2010). Fostering digital inclusion and accessibility : the porsimples project for simplification of portuguese texts. In *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, p. 46–53 : Association for Computational Linguistics.

ANDERSON R. C. & DAVISON A. (1986). Conceptual and empirical bases of readability formulas. *Center for the Study of Reading Technical Report ; no. 392*.

APROSIO A. P., TONELLI S., TURCHI M., NEGRI M. & DI GANGI M. A. (2019). Neural text simplification in low-resource conditions using weak supervision. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, p. 37–44.

BROUWERS L., BERNHARD D., LIGOZAT A.-L. & FRANÇOIS T. (2012). Simplification syntaxique de phrases pour le français (syntactic simplification for french sentences)[in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*, p. 211–224.

BROUWERS L., BERNHARD D., LIGOZAT A.-L. & FRANÇOIS T. (2014). Syntactic sentence simplification for french. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, p. 47–56.

CARROLL J., MINNEN G., CANNING Y., DEVLIN S. & TAIT J. (1998). Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, p. 7–10.

CHANDRASEKAR R., DORAN C. & SRINIVAS B. (1996). Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, p. 1041–1044 : Association for Computational Linguistics.

COSTER W. & KAUCHAK D. (2011). Learning to simplify sentences using wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, p. 1–9 : Association for Computational Linguistics.

DALE E. & CHALL J. S. (1948). A formula for predicting readability : Instructions. *Educational research bulletin*, p. 37–54.

FENG L., JANSCHE M., HUENERFAUTH M. & ELHADAD N. (2010). A comparison of features for automatic readability assessment. In *Proceedings of the 23rd international conference on computational linguistics : Posters*, p. 276–284 : Association for computational linguistics.

FLESCH R. & GOULD A. J. (1949). *The art of readable writing*, volume 8. Harper New York.

FRANÇOIS T. & FAIRON C. (2012). An ai readability formula for french as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, p. 466–477 : Association for Computational Linguistics.

FRANÇOIS T. & WATRIN P. (2011). Quel apport des unités polylexicales dans une formule de lisibilité pour le français langue étrangère. *Traitement Automatique des Langues Naturelles*, p.49̃.

GALA N., FRANÇOIS T., JAVOUREY-DREVET L. & ZIEGLER J. C. (2018). La simplification de textes, une aide à l'apprentissage de la lecture. *Langue française*, (3), 123–131.

GALA N. & ZIEGLER J. (2016). Reducing lexical complexity as a tool to increase text accessibility for children with dyslexia. In *Computational Linguistics for Linguistic Complexity, workshop at COLING (Computational Linguistics conference)*, Osaka, Japan. HAL : hal-01757941.

INUI K., FUJITA A., TAKAHASHI T., IIDA R. & IWAKURA T. (2003). Text simplification for reading assistance : a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, p. 9–16 : Association for Computational Linguistics.

KALCHBRENNER N. & BLUNSOM P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 1700–1709.

KINCAID J. P., FISHBURNE JR R. P., ROGERS R. L. & CHISSOM B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

PAETZOLD G. H. & SPECIA L. (2013). Text simplification as tree transduction. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

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

PETERSEN S. E. & OSTENDORF M. (2009). A machine learning approach to reading level assessment. *Computer speech & language*, **23**(1), 89–106.

QUINIOU S. & DAILLE B. (2018). Towards a Diagnosis of Textual Difficulties for Children with Dyslexia. In *11th International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan. HAL : hal-01737726.

SAGGION H., ŠTAJNER S., BOTT S., MILLE S., RELLO L. & DRNDAREVIC B. (2015). Making it simplext : Implementation and evaluation of a text simplification system for spanish. *ACM Transactions on Accessible Computing (TACCESS)*, **6**(4), 1–36.

SERETAN V. (2012). Acquisition of syntactic simplification rules for french.

SIDDHARTHAN A. (2006). Syntactic simplification and text cohesion. *Research on Language and Computation*, **4**(1), 77–109.

SIDDHARTHAN A. & KATSOS N. (2010). Reformulating discourse connectives for non-expert readers. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, p. 1002–1010 : Association for Computational Linguistics.

SIDDHARTHAN A. & MANDYA A. (2014). Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, p. 722–731.

SPECIA L. (2010). Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language*, p. 30–39 : Springer.

ŠTAJNER S., EVANS R., ORASAN C. & MITKOV R. (2012). What can readability measures really tell us about text complexity. In *Proceedings of the the Workshop on Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, p. 14–21 : Citeseer.

ŠTAJNER S. & SAGGION H. (2018). Data-driven text simplification. In *Proceedings of the 27th International Conference on Computational Linguistics : Tutorial Abstracts*, p. 19–23.

SU K.-Y., WU M.-W. & CHANG J.-S. (1992). A new quantitative quality measure for machine translation systems. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, p. 433–439 : Association for Computational Linguistics.

SURYA S., MISHRA A., LAHA A., JAIN P. & SANKARANARAYANAN K. (2018). Unsupervised neural text simplification. *arXiv preprint arXiv :1810.07931*.

TILLMANN C., VOGEL S., NEY H., ZUBIAGA A. & SAWAF H. (1997). Accelerated dp based search for statistical translation. In *Fifth European Conference on Speech Communication and Technology*.

VAJJALA S. & MEURERS D. (2014). Assessing the relative reading level of sentence pairs for text simplification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, p. 288–297.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

WILLIAMS S. & REITER E. (2008). Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, **14**(4), 495–525.

XU W., NAPOLES C., PAVLICK E., CHEN Q. & CALLISON-BURCH C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, **4**, 401–415.

ZHANG X. & LAPATA M. (2017). Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 584–594, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/D17-1062.

ZHU Z., BERNHARD D. & GUREVYCH I. (2010). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, p. 1353–1361 : Association for Computational Linguistics.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Évolution phonétique des langues et réseaux de neurones : travaux préliminaires

Clémentine Fourrier
Inria, 2 rue Simone Iff, 75012 Paris, France
clementine.fourrier@inria.fr

RÉSUMÉ

La prédiction de cognats est une tâche clef de la linguistique historique et présente de nombreuses similitudes avec les tâches de traduction automatique. Cependant, alors que cette seconde discipline a vu fleurir l'utilisation de méthodes neuronales, celles-ci restent largement absentes des outils utilisés en linguistique historique. Dans ce papier, nous étudions donc la performance des méthodes neuronales utilisées en traduction (les réseaux encodeur-décodeur) pour la tâche de prédiction de cognats. Nous nous intéressons notamment aux types de données utilisables pour cet apprentissage et comparons les résultats obtenus, sur différents types de données, entre des méthodes statistiques et des méthodes neuronales. Nous montrons que l'apprentissage de correspondances phonétiques n'est possible que sur des paires de cognats, et que les méthodes statistiques et neuronales semblent avoir des forces et faiblesses complémentaires quant à ce qu'elles apprennent des données.

ABSTRACT

Sound change and neural networks: preliminary experiments

Cognate prediction is a key task in historical linguistics that presents a number of similarities with machine translation. However, although neural methods are now widespread in machine translation, they are still largely unused in historical linguistics. In this paper, we study the performance of neural methods (more specifically encoder-decoder networks) for the task of cognate prediction. We focus in particular on the types of data that can be used for this task, and compare the performance of statistical and neural methods. We show that sound correspondances can only be learned using cognate datasets, and that statistical and neural methods seem to have complementary strengths and weaknesses regarding what they learn about the data.

MOTS-CLÉS : Changements phonétiques, réseaux de neurones, prédiction de cognats, linguistique historique.

KEYWORDS: Regular sound changes, neural networks, cognate prediction, historical linguistics.

1 Contexte

Osthoff *et al.* (2014) furent les premiers à formaliser la notion de régularité des changements phonétiques, pierre angulaire de la méthode comparative sur laquelle reposent toutes les études réalisées depuis en phonétique historique. Leur observation empirique était la suivante : une transformation diachronique d'un phone en un autre phone, dans un contexte donné, est régulière et sans exception ; à ce titre, tous les mots contenant ce même phone dans ce même contexte vont eux aussi subir cette transformation phonétique. Cette observation fut rendue possible par la comparaison des représenta-

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 110–122.
HAL : hal-02786192.

tions phonétisées d'ensembles de mots appelés cognats [1], qui permit d'identifier des motifs récurrents dans les correspondances phonétiques.

L'utilisation itérative de cette méthode pour, successivement, identifier de nouvelles règles de changement phonétique en comparant des cognats, puis trouver des cognats inédits grâce aux règles nouvellement identifiées, constitue le cœur de la phonétique historique, et permet de définir deux de ses tâches principales : l'identification de cognats et la prédiction de changements phonétiques.

Avec l'émergence des méthodes informatiques, de nouvelles façons de traiter ces tâches, s'appuyant sur le traitement massif de nombreuses données, virent le jour. Une grande majorité d'entre elles reposent sur des comparaisons automatiques de lexèmes, qui combinent des méthodes d'alignement phonétique, soit avec des calculs de distances de type Levenshtein ou Turchin, soit avec des méthodes d'agrégation (List *et al.* (2017) font une comparaison de ces différents méthodes). Plus récemment, certains travaux commencèrent à capitaliser sur les fortes similitudes entre les tâches précédemment décrites et la traduction automatique : les deux visent à apprendre le passage d'une séquence d'éléments ordonnés à une autre séquence d'éléments ordonnés (problème « *many to many* »). Dans cette optique, Beinborn *et al.* (2013) utilisent Moses, logiciel appliquant des méthodes de traduction statistique (Koehn *et al.*, 2007), pour de la prédiction de cognats (avec une définition large de cognat).

Cependant, le domaine de la traduction automatique s'est maintenant enrichi de méthodes neuronales, et, au vu de leur prééminence dans la discipline, on est en droit de se demander pourquoi ces méthodes n'ont quasiment jamais été appliquées à l'heure actuelle pour les problématiques de linguistique historique, notamment en ce qui concerne le modèle de référence de la traduction : l'encodeur-décodeur. A notre connaissance, le seul travail de recherche réalisé pour l'instant sur l'application de cette architecture à ces problématiques est le rapport de mémoire de Dekker (2018), dont les conclusions sont peu reluisantes : pour la tâche de prédiction de cognats, un simple perceptron surpasse les réseaux récurrents de type encodeur-décodeur (en utilisant 1016 formes phonétisées appartenant à une centaine de langues).

Ces travaux ont donc soulevé la problématique suivante : est-il possible d'utiliser des réseaux de neurones de type encodeur-décodeur pour apprendre des correspondances phonétiques, et si oui, sur quel type de données ?

Nous tentons de répondre à ces questions, avec quelques expériences préliminaires. En premier lieu, nous étudions les types de données sur lesquelles il est possible d'apprendre des correspondances phonétiques, et notamment s'il est possible de pallier le faible nombre de données de cognats en utilisant des lexiques bilingues. La seconde expérience s'intéresse à la pertinence de l'utilisation et au paramétrage de réseaux de neurones pour cette tâche. Enfin, les dernières expériences visent à étendre, de façon préliminaire, ces résultats à des données réelles.

1. Des cognats sont des mots de langues différentes, descendants d'un même « mot-ancêtre » commun, celui-ci appartenant à une langue parente de toutes les langues concernées. Ces mots ont donc vécu toutes les évolutions phonétiques de leurs langues respectives. Par exemple, les mots polonais *być* 'être', tchèque *být* 'id.' et lituanien *būti* 'id.' sont tous cognats, et descendent d'un ancêtre commun en proto-balto-slave.

2 Cadre expérimental

2.1 Tâche à résoudre

Dans cet article, nous désirons étudier si un modèle neuronal peut apprendre des correspondances phonétiques. Nous définissons à cet effet la tâche d'intérêt comme étant la « traduction » de cognats phonétisés d'une langue vers une autre.

Cette tâche n'est pas triviale, et ce pour plusieurs raisons. Déjà, les données de cognats sont rares. Les jeux de cognats font en général de 100 à 200 mots, ce qui est très peu pour apprendre avec des méthodes neuronales sans sur-apprentissage (ces méthodes comprenant elles-même plusieurs centaines de poids mathématiques à ajuster). Ensuite, c'est une tâche qui peut présenter des ambiguïtés importantes, selon la direction de prédiction étudiée. Si l'on va d'une langue mère vers sa langue fille, les changements phonétiques sont strictement réguliers, et à un mot de la langue mère est associé un seul mot de la langue fille, sans ambiguïtés. Mais la transformation inverse n'est pas évidente : un mot de la langue fille pourrait, formellement, descendre de plusieurs mots de la langue mère [2]. Pour cette raison, nous demanderons à nos modèles de produire plusieurs prédictions, de une à trois, et nous comparerons l'impact du nombre de réponse prédites sur la précision. Enfin, les données qui sont accessibles sont généralement bruitées, et à ce titre, présentent un défi supplémentaire : en apprendre les correspondances phonétiques sans en apprendre le bruit.

2.2 Modèle étudié : modèle neuronal (MEDeA)

L'architecture neuronale que nous utilisons est un des modèles de référence en traduction automatique : l'encodeur-décodeur (Sutskever *et al.*, 2014) avec attention (Bahdanau *et al.*, 2015; Luong *et al.*, 2015). Ce type de modèle transforme l'entrée (une séquence ordonnée, de phones dans notre cas, de mots en traduction automatique) en une représentation intermédiaire vectorielle (« *hidden representation* ») grâce à l'encodeur, une succession de réseaux de neurones récurrents. Cette représentation intermédiaire est ensuite lue par le décodeur, qui prédit chaque phone (resp. mot) de la séquence finale successivement (dans la langue de sortie) en fonction de l'enchaînement des phones (resp. mots) précédemment prédits. Pour cet article, nous utilisons notre implémentation de cet algorithme, MEDeA (*Multiway Encoder Decoder Architecture*, en Python et PyTorch), qui utilise un encodeur différent pour chaque langue d'entrée, et un décodeur indépendant, avec sa propre attention, et différent pour chaque langue de sortie. Pour contraindre la représentation intermédiaire à un seul espace, le réseau apprend sur toutes les paires de langues possibles (y compris d'une langue vers elle-même). Notre architecture permet de passer d'un grand nombre de langues à un grand nombre de langues identiques ou non.

2. Par exemple, considérons un son [son1], dans la langue mère, qui évolue régulièrement en [son2] dans sa langue fille, tandis que [son2] dans la langue mère reste [son2] dans la langue fille. Pour passer de la langue mère à la langue fille, la situation ne présente pas d'ambiguïté : [son1] devient [son2], et [son2] devient [son2]. Par contre, à l'inverse, si on a [son2] dans la langue fille, il peut correspondre à [son1] comme à [son2] dans la langue mère.
Prenons un exemple concret. Dans certains cas, le [b] latin évolue en un [v] en italien, comme pour [kaˈbal.lus], 'cheval' qui devient en italien [kaˈvaːlo] 'id.'. Cependant, le son [v] latin reste un [v] en italien, comme dans [vita] 'vie', identique dans les deux langues. Un [v] italien correspond-il à un [v] ou à un [b] initial en latin ? Remonter de la langue fille à la langue mère présente ici une ambiguïté.

2.3 Modèle de référence : modèle statistique (Moses)

Moses est le modèle statistique de référence en traduction automatique statistique (Koehn *et al.*, 2007). Il fonctionne en deux étapes. Pour la première, l'entraînement, les données bilingues sont tokenisées et alignées avec l'aide de GIZA++ (Och & Ney, 2003), puis servent à l'apprentissage d'un modèle de langue tri-gramme de la langue de sortie, d'un tableau de correspondances entre les phones de la langue d'entrée et de la langue de sortie, et un modèle de réorganisation pour gérer l'ordre des phones. Lors de la seconde étape, la mise au point (« *fine tuning* »), les poids respectifs de chacun de ces modèles dans le système global sont ajustés grâce à un jeu de données de développement. Il est à noter qu'une différence notable entre le modèle neuronal et le modèle statistique est que le second ne peut apprendre que la traduction d'une langue vers une autre, là où le modèle neuronal apprend la correspondance de plusieurs langues en même temps.

2.4 Métrique d'évaluation : BLEU

Notre métrique d'évaluation est BLEU (Papineni *et al.*, 2002), plus spécifiquement l'implémentation SacreBLEU (Post, 2018). Le score BLEU calcule le pourcentage d'éléments communs (mots en traduction automatique, phones dans notre cas) entre la séquence de départ et celle d'arrivée, des unigrammes aux quadrigrammes d'éléments. La critique usuelle de cette métrique est qu'elle tend à mal noter des traductions correctes mais non présentes dans le jeu de référence ; cette critique ne s'applique pas à nos expériences, dans la mesure où il n'existe qu'une seule « traduction » possible d'un cognat en son équivalent dans une autre langue.

3 En l'absence de jeux de cognats, les lexiques bilingues peuvent-ils être utilisés pour étudier l'évolution phonétique des langues ?

3.1 Données

Notre tâche est définie comme l'apprentissage, à partir de paires de cognats, de régularités phonétiques (issues des changements phonétiques réguliers que ces mots ont vécu dans leurs langues respectives). Cependant, les jeux de cognats, des plus classiques comme les listes de Swadesh (1955) à des versions plus récentes comme les initiatives de Dunn (2012) ou Dunn *et al.* (2016), sont de taille trop restreinte (d'une cinquantaine à quelques centaines de paires de mots) pour entraîner des réseaux de neurones. Nous avons donc décidé d'explorer l'utilisation de lexiques bilingues génériques, qui, pour des langues proches, incluent forcément des paires de cognats (en un nombre possiblement plus important que les jeux de référence [3]) au sein d'une majorité de paires qui, ne l'étant pas, constituent du bruit pour notre tâche.

Pour ces expériences préliminaires, nous nous sommes donc intéressés à deux types de données : des

3. Les jeux de références contiennent, au grand maximum, quelques centaines de paires de mots comme cognats attestés. On peut supposer que dans un lexique bilingue d'au moins une dizaine de milliers de mots se trouvent, en plus de ceux-ci, des cognats non attestés pour le moment, mais qui, par leur nature, contiennent les changements phonétiques que l'on cherche à étudier. Ce nombre est malheureusement difficilement quantifiable.

Langues	PL-CZ	PL-LT	PL-IT
Jeu d'entrainement (lexique bilingue)	13,216	6,290	17,158
Jeu de test (paires de cognats)	370	47	57

TABLE 1 – Nombre de paires de mots par jeu de données

jeux de cognats (jeux de test), et des lexiques bilingues (jeux d'entraînement), entre du polonais (PL) et, de la langue la plus proche à la moins proche, du tchèque (CZ), du lituanien (LT), et de l'italien (IT).

3.1.1 Extraction et pré-traitement

EtymDB (Sagot, 2017) est une base de données étymologique extraite automatiquement du Wiktionary, comprenant des lexèmes (triplets de la forme ⟨langue, lemme, sens représenté par une ou plusieurs gloses en anglais⟩) reliés par différentes relations étymologiques typées, dont la relation « hérité de ». Pour générer les paires de cognats pour nos premières expériences, nous suivons les chemins entre les mots, considérant que deux d'entre eux sont cognats si jamais ils partagent un ancêtre dans une de leurs langues parentes communes, et en descendent en ligne droite [4].

YaMTG (Hanoka & Sagot, 2014) est un graphe de traduction multilingue et libre extrait automatiquement du Wiktionary, et une des rares bases de données libres contenant nos langues d'intérêt. Nous en extrayons les entrées bilingues pour nos trois paires de langues, pour créer trois lexiques bilingues (après en avoir retiré les paires de mots contenant des caractères inattendus).

Ces deux jeux sont ensuite phonétisés avec Espeak (Duddington, 2015). Dans le cas où des paires de mots contiennent un lexème identique à l'entrée mais des traductions différentes dans la langue de sortie, ne sont conservées que les paires avec la distance de Levenshtein la plus courte (méthode permettant le meilleur rappel de cognat d'après List *et al.* (2017)).

3.1.2 Propriétés

Les jeux d'entraînement résultants (Table 1) comprennent environ 13 000 paires entre le polonais et le tchèque, 6 000 entre le polonais et le lituanien, et 17 000 entre le polonais et l'italien. Les jeux de cognats sont en moyenne 100 fois plus petits : 370 paires entre le polonais et le tchèque, contre seulement 47 entre le polonais et le lituanien, et 57 entre le polonais et l'italien.

4. Les ancêtres communs présents dans la base sont le slave commun, le proto-balto-slave et le proto-indo-européen pour le polonais et le tchèque, les deux derniers pour le polonais et le lituanien, et seulement le proto-indo-européen pour le polonais et l'italien.

3.2 Paramètres expérimentaux

Après des expériences préliminaires, nous avons déterminé que les meilleurs paramètres pour le modèle neuronal, dans le cadre de cette expérience, étaient l'utilisation d'une couche cachée de type Gated Recurrent Units (GRU) de dimension 100 pour l'encodeur et le décodeur. Les décodeurs utilisent de plus l'attention « dot » de Luong (Luong *et al.*, 2015). L'optimiseur est de type Adam, avec un taux d'apprentissage de 0,001.

Pour le modèle statistique, les paramètres utilisés sont ceux décrits précédemment.

Le jeu d'entraînement est séparé en 70%-30%, 80%-20%, et 90%-10% pour les étapes, respectivement, d'apprentissage et de mise au point. Chaque séparation est aléatoirement générée 10 fois, de façon à observer l'impact de la variation des tailles respectives de ces deux jeux sur l'entraînement. Dans notre cas, cette séparation n'ayant eu aucun impact statistiquement significatif sur les résultats, ceux-ci sont tous traités ensemble ici.

3.3 Résultats

LANGUES	PL→CZ	PL→LT	PL→IT
	Résultats sur les jeux de tests		
MEDeA	48.44 ± 1.13	19.22 ± 0.99	26.75 ± 1.05
Moses	54.43 ± 0.41	20.55 ± 0.62	28.37 ± 0.62

TABLE 2 – Scores BLEU de MEDeA et Moses (moyenne sur 30 expériences).

En observant la Table 2, le premier constat que nous faisons est que tous nos résultats sont masqués par une contrainte matérielle : la quantité de données d'entraînement. En effet, le jeu PL-LT est deux à trois fois plus petit que ses confrères, et il est le jeu avec les plus mauvais résultats. Cependant, si seule la taille des données avait un impact sur l'apprentissage, on s'attendrait à ce que le jeu le mieux prédit soit celui entre l'italien et le polonais, soit le plus gros jeu, alors que les meilleurs résultats sont ceux obtenus entre le polonais et le tchèque, de 20 points BLEU meilleurs pour un jeu d'entraînement 30% plus petit. Nous supposons donc que, sous réserve d'avoir suffisamment de données, plus des langues sont proches, plus leurs correspondances sont faciles à apprendre.

Le second constat est que la méthode neuronale est systématiquement moins bonne que la méthode statistique sur la tâche de prédiction de cognats après un entraînement sur des lexiques bilingues. Ceci peut indiquer soit que les méthodes neuronales de type encodeur-décodeur ne sont pas adaptées à cette tâche dans l'absolu, soit que les méthodes statistiques ont une meilleure capacité à extraire des informations de données très bruitées. Nous notons également que les résultats ne sont en moyenne pas très bons, et que la piste des lexiques bilingues ne semble pas être aussi intéressante qu'elle aurait pu. Pour différencier entre ces hypothèses, nous proposons les expériences suivantes.

4 Les méthodes neuronales conviennent-elles à l'apprentissage de correspondances phonétiques ? Expériences sur des données artificielles

4.1 Contexte

L'utilisation de lexiques bilingues pour apprendre à prédire des cognats s'est avérée être une piste peu satisfaisante. Cependant, deux raisons majeures peuvent être à l'origine des difficultés rencontrées par nos modèles : soit les réseaux de neurones ne sont pas aussi bien adaptés que les méthodes statistiques à l'apprentissage de ces changements, soit ils sont plus sensibles au bruit que des méthodes statistiques, et les lexiques bilingues constituaient des données trop bruitées pour l'apprentissage de notre tâche.

Il est possible d'invalider facilement une des hypothèses évoquées précédemment : si les réseaux de neurones ne conviennent pas à l'apprentissage des changements phonétiques, alors ils n'apprendront jamais aussi bien ou mieux que des méthodes statistiques, même sur des données parfaites. Par contre, s'ils sont capables d'apprendre sur des données parfaites, alors le problème vient plus probablement de leur sensibilité au bruit que de leur inadéquation à la tâche.

Pour comprendre ce qu'il est effectivement possible d'apprendre ou non, nous décidons de générer des lexiques phonétiques artificiels, composé d'une proto-langue, et de deux langues filles générées en appliquant à la langue mère des changements phonétiques réguliers. Cette méthode présente deux avantages : elle permet d'étudier la taille minimale de données nécessaires pour apprendre, et de maîtriser complètement les paramètres liés aux données elles-mêmes, de la richesse à la quantité de bruit. Cependant, il est important que les données, bien qu'artificielles, obéissent à des règles de construction et d'évolution réalistes, pour que les résultats des expériences puissent être transposables à des données réelles.

4.2 Données artificielles

Nous choisissons pour cela de créer une proto-langue à partir d'un inventaire de phones et d'une phonotactique (organisation syllabique des sons dans la langue), puis d'en dériver les langues filles à partir de l'application séquentielle de changements phonétiques plausibles.

Nous avons ainsi développé un algorithme, qui, à partir d'un inventaire de phones et d'une phonotactique, génère un lexique d'une taille choisie. Dans le cadre de ces expériences, nous choisissons de nous inspirer du latin pour la proto-langue, et des langues romanes pour ses langues filles. Nous utilisons donc trois sources historiques. Pour la génération de la proto-langue (PL) nous utilisons tout d'abord l'inventaire phonétique des langues romanes : chaque lexique généré en utilise les phones communs à toutes les langues et tire aléatoirement un sous-ensemble des phones moins courants. Nous utilisons ensuite une version simplifiée de la phonotactique du latin classique (inspirée de (Cser, 2016)), qui nous permet de générer des mots à partir d'un nombre de syllabes aléatoirement choisi, lesquelles sont construites en suivant les règles phonotactiques liées à leur emplacement au sein des mots. Enfin, la dernière source sont les changements phonétiques des langues romanes. Pour générer les langues filles, l'algorithme choisit aléatoirement un sous-ensemble de changements phonétiques parmi ceux possibles (dont l'apocope, l'épenthèse, la palatalisation, la lénition, la prothèse de voyelles et la diphtongaison), puis les applique successivement au lexique de la proto-langue pour générer le

lexique d'une langue fille.

Pour nos expériences, nous avons finalement généré, à partir d'une proto-langue (PL), deux langues filles (F1 et F2) avec 15 changements phonétiques chacune, soit 20 000 triplets de mots. Voici deux exemple du type de données phonétisées obtenues : $[\text{stra}]_{\text{PL}} > [\text{isdre}]_{\text{F1}}$, $[\text{estre}]_{\text{F2}}$ et $[\text{ʒolpast}]_{\text{PL}} > [\text{ʒolbes}]_{\text{F1}}$, $[\text{ʒolpes}]_{\text{F2}}$

4.3 Paramètres expérimentaux

Notre but lors de ces expériences est double : déjà, vérifier si les méthodes neuronales peuvent apprendre, mais également, le cas échéant, déterminer si la quantité de données a un impact sur les performances relatives des deux types de modèles. Pour cette raison, nous réitérons les expériences pour différentes tailles de jeux de données : 500, 1000, 1500, 2000 et 3000 triplets de mots, divisées en 80% pour l'entraînement et 20% pour le test. Ces données sont choisies aléatoirement parmi les 20 000 triplets générés précédemment, selon 3 graines d'aléa (« *random seeds* ») différentes.

En ce qui concerne les modèles statistiques, 80% du jeu d'entraînement est utilisé pour leurs étapes d'apprentissage et 20% pour leurs étapes de mise au point ; un modèle statistique différent doit être entraîné par paire de langue possible.

Le modèle neuronal, quand à lui, utilise toutes les données d'entraînement pour l'apprentissage, de toutes les langues à toutes les langues en une seule fois ; la taille de couche cachée donnant les meilleurs résultat était de 25 après des expériences préliminaires [5].

Les deux modèles prédisent de la meilleure aux trois meilleures réponses.

4.4 Résultats

Sur les données synthétiques dénuées de bruit, le modèle statistique comme le modèle neuronal apprennent très bien à prédire d'une langue fille (F1) à une autre langue fille (F2), et atteignent des scores très nettement supérieurs à ceux obtenus lors des expériences préliminaires sur les lexiques bilingues : entre 88 et 97 BLEU pour MEDeA et 90 à 96 pour Moses, soit des résultats équivalents, comme on peut le voir sur les colonnes F1-F2 et F2-F1 de la figure 1. Les réseaux de neurones peuvent donc apprendre à prédire des cognats d'une langue fille à une autre langue fille, à partir de données de cognats, si tant est que ces données soient de qualité et de quantité suffisante.

4.4.1 Impact de la direction de prédiction sur les résultats

Cependant, nous notons également que toutes les directions de prédictions ne sont pas équivalentes. Prédire de la proto-langue (PL) à une langue fille (F) donne les meilleurs résultats (de 94 à 99 BLEU), tandis que prédire d'une langue fille à la langue mère est, de très loin, la tâche la plus difficile (de 60 à 80 BLEU). De plus, prédire la deuxième et la troisième meilleure réponse augmente considérablement les scores BLEU dans le cas des situations présentant une ambiguïté forte (F→PL), et ce quel que soit le modèle considéré.

5. La différence considérable de taille de la couche cachée entre cette expérience et la précédente s'explique par la différence de taille entre les données utilisées, d'un facteur de 3 à 20.

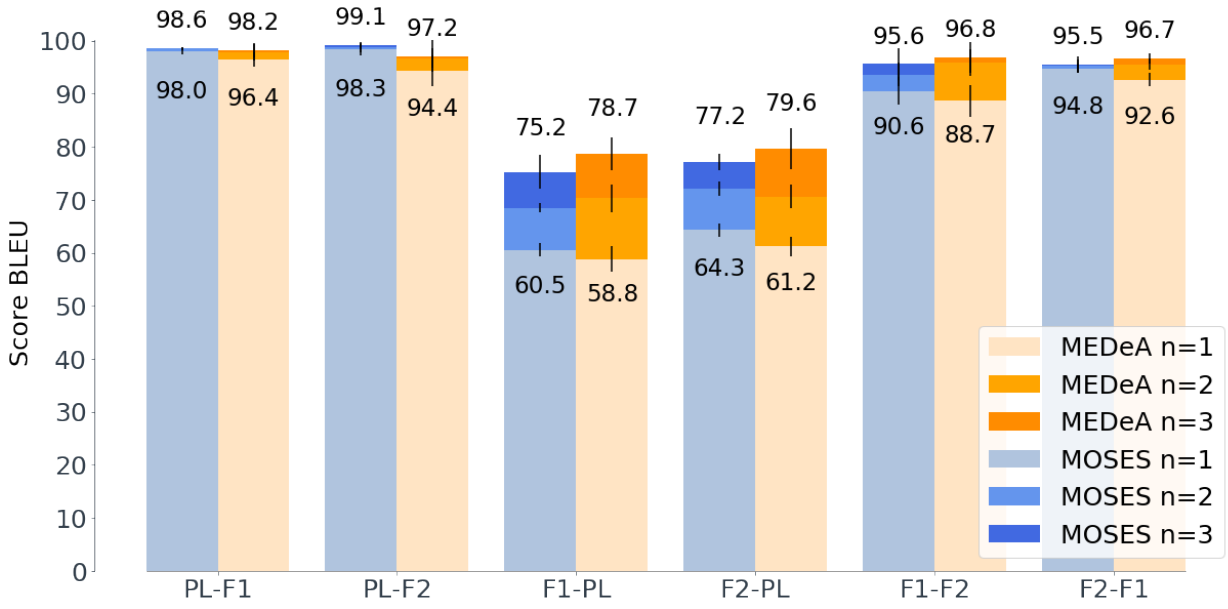
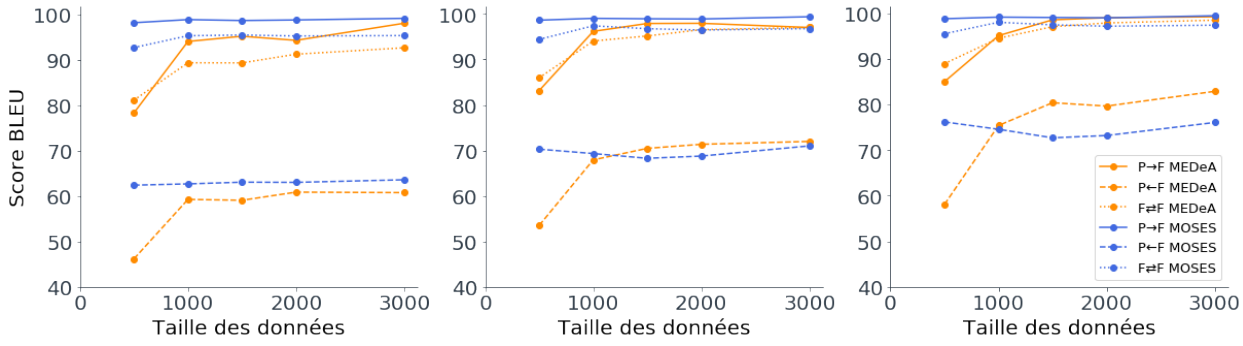


FIGURE 1 – Scores BLEU pour les n meilleures prédictions, à partir de 1000 paires de mots (en fonction de la direction de prédiction).

4.4.2 Impact de la taille des données et du nombre de prédictions sur les résultats



(a) n=1 (b) n=2 (c) n=3

FIGURE 2 – Scores BLEU pour les n meilleures prédictions.

Le modèle statistique est systématiquement meilleur que le modèle neuronal quand on calcule le score BLEU sur la meilleure réponse seulement, encore que les performances se rejoignent quand on dépasse 2000 triplets de mots pendant l'entraînement, comme on peut le voir dans la figure 2.

Par contre, pour les deux ou trois meilleures réponses, les résultats changent : le modèle statistique est significativement meilleur pour de très petites tailles de données (500 triplets) [6], mais le modèle neuronal est bien meilleur pour un grand nombre de données (2000 triplets et plus). Sur la zone intermédiaire, entre 1000 et 1500 triplets, les performances sont à peu près équivalentes entre les modèles.

Cette expérience nous a donc permis de montrer que, pour des données idéales, un modèle neuronal

6. On suppose que le réseau de neurone est moins performant pour de petites tailles de données car il sur-apprend.

est aussi performant qu'un modèle statistique à partir de 1500 paires de mots par couple de langue, notamment quand on s'intéresse aux situations impliquant de l'ambiguïté. Qu'en est-il sur des données réelles ?

5 Peut-on apprendre des changements phonétiques sur des données réelles ?

Pour déterminer si nos résultats sont généralisables, nous menons les mêmes expériences sur des données réelles. Il est attendu que ces expériences fonctionneront moins bien que les précédentes, dans la mesure où les données réelles contiennent du bruit, complètement absent de nos données artificielles.

5.1 Données réelles

Nous avons besoin de travailler sur un jeu de cognats de langues réelles liées impliquant une langue parente bien connue et plusieurs langues filles proches mais différentes. Nous choisissons d'étudier le latin (LA) comme langue mère et l'italien (IT) et l'espagnol (ES) comme langues filles.

Pour extraire nos jeux de données, nous utilisons EtymDB 2.0 (Fourrier & Sagot, 2020), la version la plus récente de la base de données EtymDB présentée à la section 3.1.1. Nous utilisons le même raisonnement pour extraire les cognats : 2 mots sont cognats s'ils partagent un ancêtre pour une de leurs langues parentes communes (ici, le latin, le latin pré-classique, le proto-italique et le proto-indo-européen pour les paires LA-IT et LA-ES, et ces langues plus le latin vulgaire pour la paire IT-ES). La phonétisation et le pré-traitement sont de nouveau effectués avec Espeak puis l'application d'une distance de Levensthein pour retirer les doublons erronés. Toute séquence de phone est précédée d'un token de début de phrase indiquant sa langue, et terminée par un token de fin de phrase.

Le jeu final contient 605 triplets LA-ES-IT, qui ont été manuellement ré-éxaminés, et serviront principalement de jeu de test (2/3 de test, 1/3 de train). Les données d'entraînement sont constituées de 5040 paires de cognats pour IT-LA, 4208 pour ES-LA, et 1801 pour ES-IT, desquelles sont retirées les données de test.

5.2 Paramètres expérimentaux

MEDeA est entraîné avec toutes les données sur toutes les combinaisons possibles entre les langues (IT, ES, LA), pendant 50 itérations (« *epochs* »). Nous l'entraînons sur 3 graines d'aléa, et comparons des tailles de couche cachée entre 12 et 50, et la précision de la meilleure aux trois meilleures prédictions. Après observations, nous constatons que pour ces jeux, la meilleure taille de couche cachée est de 37, avec des plongements vectoriels de taille 10, et ce sont les valeurs que nous utiliserons pour la suite des expériences.

Moses est entraîné sur les différentes combinaisons de paires de langues séparément, avec les mêmes divisions dans les données. Les données triples sont traitées comme des combinaisons de paires.

5.3 Résultats préliminaires

Nous observons, durant ces expériences, que le modèle statistique est systématiquement meilleur que le modèle neuronal, d'environ 15 points. Les réseaux de neurones sont très sensibles au bruit et aux incohérences dans les données.

Cependant, là où le modèle statistique obtient pour l'instant une performance absolue supérieure, le modèle neuronal semble apprendre une structure sous-jacente des données, qui lui permet de mieux gérer les cas d'ambiguïté. En effet, pour chacune des paires de mots de nos jeux de test, nous associons à une entrée unique une sortie unique. Ainsi, quand un modèle prédit plusieurs réponses, une seule sera correcte *par rapport au jeu de données*. Cependant, les autres ne seront pas pour autant fausses dans l'absolu, et on peut distinguer 3 cas :

1. le modèle prédit des réponses secondaires historiquement valides (par exemple, le même adjectif, accordé différemment)
2. le modèle prédit des réponses secondaires historiquement plausibles (par exemple, un nom dans un autre genre, incorrect mais plausible par rapport à la structure de la langue)
3. le modèle prédit des réponses secondaires complètement incorrectes

Observons quelques exemples de l'italien au latin. Le réseau de neurone prédit à plusieurs reprises des formes grammaticalement valides, comme [rustiko] 'rustique', venant de [rʊstɪkʊs] 'de la campagne', qui voit son ancêtre prédit comme étant [rʊstɪkʊs] (masc. - bonne réponse), [rʊstɪkʊm] (neut. — cas 1), ou [rʊstɪkss] (aucun sens — cas 3) par MEDeA, contre [rʊkʊstɪ], [rʊɪkɔst] ou [ʊsrtɪkwʊs], trois formes dépouvues de sens, par Moses (toutes cas 3). MEDeA nous a d'ailleurs permis d'identifier des erreurs dans nos jeux de données : [ramo] 'branche' $<$ [ramʊs] 'branche', était relié de façon erronée à [radɪks] 'racine' (qui est un cognat de [ramʊs]) ; MEDeA a prédit cet ancêtre comme étant [ramʊs] (masc. — bonne réponse), [ramʊ] (cas 3), ou [ramʊm] (forme neutre du nom, incorrecte, mais plausible, soit le cas 2), tandis que Moses a prédit [mʊr], [rɛam], ou [raɛm] (toutes le cas 3 à nouveau).

Le modèle statistique produit donc plus souvent *la* bonne réponse par rapport à nos données, obtenant ainsi un meilleur score, là où le modèle neuronal produit plus souvent plusieurs réponses plausibles.

6 Conclusion

Dans cet article, nous avons d'abord montré que les lexiques bilingues, bien que semblant au premier abord convenir à la tâche d'apprentissage de changements phonétiques (car porteurs de cette information et de taille raisonnable), étaient en réalité trop bruités, pour les réseaux de neurones comme les méthodes statistiques étudiées. Nous avons ensuite montré, en travaillant sur des données artificielles, que ces deux types d'algorithmes présentent des forces et faiblesses complémentaires sur des jeux de taille réaliste sans être trop restreints ; les méthodes statistiques sont meilleures dans les cas ne présentant pas d'ambiguïté (direction de prédiction chronologique), et les réseaux de neurones dans les cas en présentant beaucoup (direction de prédiction chronologique inverse). Enfin, en cherchant à confirmer ces résultats sur des données réelles, nous avons montré que les méthodes statistiques sont, avec les paramètres choisis pour ces expériences, plus performantes que les méthodes neuronales, mais que ces dernières semblent faire de meilleures généralisations sur les données. Des expériences complémentaires restent à faire. Une première étape serait la création des données artificielles bruitées, ou subissant des changements phonétiques plus complexes, pour

étudier la performance respective des deux types de modèles sur celles-ci. Une seconde est l'étude plus détaillée des apprentissages des différents modèles, à l'échelle du mot et du phone ; pour ce faire, il pourrait être intéressant de chercher une métrique plus pertinente pour la tâche de linguistique historique que le score BLEU, qui pénaliserait moins les phones prédits quand ils sont proches de ceux attendus [7].

Références

BAHDANAU D., CHO K. & BENGIO Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

BEINBORN L., ZESCH T. & GUREVYCH I. (2013). Cognate production using character-based machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, p. 883–891, Nagoya, Japan : Asian Federation of Natural Language Processing.

CSER A. (2016). *Aspects of the phonology and morphology of Classical Latin*. Thèse de doctorat, Pázmány Péter Katolikus Egyetem. DOI : 10.1111/1467-968X.12184.

DEKKER P. (2018). Reconstructing language ancestry by performing word prediction with neural networks. *Master. Amsterdam : University of Amsterdam*. DOI : 10.13140/RG.2.2.32990.33601.

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

DUDDINGTON J. (2015). eSpeak : Text to speech. http://espeak.sourceforge.net/index.html.

DUNN M. (2012). IELex : Indo-European Lexical cognacy database. http://ielex.mpi.nl/.

DUNN M., GARGETT A., RUNGE J. & KHAIT I. (2016). CoBL : Cognacy in Basic Lexicon. https://github.com/lingdb/CoBL-public.

FOURRIER C. & SAGOT B. (2020). Methodological Aspects of Developing and Managing an Etymological Lexical Resource : Introducing EtymDB-2.0. In *Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Marseilles, France.

HANOKA V. & SAGOT B. (2014). YaMTG : An Open-Source Heavily Multilingual Translation Graph Extracted from Wiktionaries and Parallel Corpora. In *Language Resources and Evaluation Conference*, Reykjavik, Iceland : European Language Resources Association. HAL : hal-01022306.

KOEHN P., HOANG H., BIRCH A., CALLISON-BURCH C., FEDERICO M., BERTOLDI N., COWAN B., SHEN W., MORAN C., ZENS R. *et al.* (2007). Moses : Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, p. 177–180. DOI : 10.5555/1557769.1557821.

LIST J.-M., GREENHILL S. J. & GRAY R. D. (2017). The potential of automatic word comparison for historical linguistics. *PLOS ONE*, **12**(1), 1–18. DOI : 10.1371/journal.pone.0170046.

LUONG T., PHAM H. & MANNING C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1412–1421, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1166.

7. Prédire une autre voyelle à la place d'un [a] mérite d'être moins pénalisé que de prédire une consonne, par exemple.

OCH F. J. & NEY H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, **29**(1), 19–51. DOI : 10.1162/089120103321337421.

OSTHOFF H., OSTHOFF H. & BRUGMANN K. (2014). In *Morphologische Untersuchungen auf dem Gebiete der indogermanischen Sprachen (reprinted)*, volume 4 de *Cambridge Library Collection - Linguistics*, p. 418–418. Cambridge University Press. DOI : 10.1017/CBO9781139600132.006.

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

POST M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation : Research Papers*, p. 186–191, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-6319.

SAGOT B. (2017). Extracting an Etymological Database from Wiktionary. In *Electronic Lexicography in the 21st century (eLex 2017)*, p. 716–728. HAL : hal-01592061.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*. DOI : 10.5555/2969033.2969173.

SWADESH M. (1955). Towards greater accuracy in lexicostatistic dating. *International Journal of American Linguistics*, **21**(2), 121–137. DOI : 10.1086/464321.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Comparing PTB and UD information for PDTB discourse connective identification

Kelvin Han[1] Phyllicia Leavitt[1] Srilakshmi Balard[1]
(1) IDMC, Université de Lorraine, Pôle Herbert Simon, 13 Rue Michel Ney, 54000 Nancy, France

RÉSUMÉ

Dans cet article, nous montrons que l'information syntaxique issue du schéma Universal Dependencies (UD) constitue une alternative viable à celle issue du schéma Penn Treebank (PTB) pour la tâche d'identification automatique des connecteurs discursifs dans le corpus du Penn Discourse Treebank. De fait, nous obtenons même des améliorations en termes de performance en utilisant des informations UD prédites par rapport à l'utilisation d'information gold PTB. Ces dernières sont traditionnellement utilisées pour cette tâche mais il existe aujourd'hui des corpus au schéma UD avec davantage de langages que le format PTB. Nos résultats sont donc prometteurs pour de futurs travaux en analyse discursive automatique multilingue ainsi que pour des applications dans un cadre réaliste où des informations PTB gold ne sont pas disponibles.

ABSTRACT

Our work on the automatic detection of English discourse connectives in the Penn Discourse Treebank (PDTB) shows that syntactic information from the Universal Dependencies (UD) framework is a viable alternative to that from the Penn Treebank (PTB) framework. In fact, we found minor increases when comparing between the use of gold standard PTB part-of-speech (POS) tag information and automatically parsed UD information. The former has traditionally been used for the task but there are now much more UD corpora and in many more languages than that available in the PTB framework. As such, this finding is promising for areas in discourse parsing such as in multilingual as well as under production settings, where gold standard PTB information may be scarce.

MOTS-CLÉS : analyse discursive automatique, Universal Dependencies, identification automatique des connecteurs discursifs.

KEYWORDS: discourse parsing, Universal Dependencies, discourse connective identification.

1 Introduction

Discourse analysis is about identifying the semantico-pragmatic links between parts of a document in order to reveal a structure that organizes a given document. This enables inferences to be made about the content of the document. Within a document, each unit (termed a 'discourse unit') is a span of text ; and its meaning depends on the meaning of its surrounding units, as well as the relation that holds between them. The presence of different types of relations are frequently marked by a specific set of wordforms (termed as 'discourse connectives'). For instance, 'because' is one of the markers for an expansion-reason relation, where one discourse unit serves to explain the cause for the other

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 123–136.
HAL : hal-02786193.

unit it is linked to. In natural language processing, discourse parsing corresponds to several tasks, the very first one being the identification of such discourse connectives. Therefore errors in it will cascade to later tasks and impact the overall performance of a discourse parser.

Identifying discourse connectives in a text is more complex than a simple search and find. This is because a connective may be lexicalized by different wordforms [1], as well as could have a non-discourse reading. Take the following sentences with the same wordform 'when' in each of them :

(1) a. I was happy **when** Michele told me she was on her way.
b. She said she would arrive, but she never told me *when*.

In sentence (1-a), 'when' is a conjunction between two verb phrases (VPs) and serves as a discourse connective ; it marks a relation of temporal succession —from the second VP to the first VP. However, in sentence (1-b), 'when' is not serving as a discourse connective. Syntactic information can be useful to distinguish between such instances of discourse and non-discourse usage ; although 'when' has a 'WHADVP' syntactic category in both sentences, it is only in the first example that it links two VPs.

Our work focuses on detecting discourse connectives automatically, since improving and avoiding errors at this step is crucial for a parser's performance ; we leave for future work the study of the other parts of the pipeline. This task is generally done on the Penn Discourse Treebank (PDTB) (Prasad *et al.*, 2008), the largest corpus annotated for discourse relations in English, and has typically been solved by training a classifier using lexical, morpho-syntactic as well as syntactic information (Pitler & Nenkova, 2009). In particular, part-of-speech (POS) tags and syntactic trees from the Penn Treebank (PTB) (Marcus *et al.*, 1993) have been used.

However, the recent release of the Universal Dependencies (UD, (Nivre *et al.*, 2016)) framework is seeing much more corpora and POS taggers made available for UD than there is available for the PTB ; they are also available in comparably many more languages in UD now. It is thus crucial to understand whether the information captured within the UD framework is sufficient for the task of automatically detecting connectives, as it would enable the development of discourse parsing systems for new languages, especially those that are not currently served by PTB-styled corpora and tools.

In our work, we seek to establish the effects of using the POS tagset from UD for the task, instead of those from the PTB. There are however, important differences between the UD and PTB frameworks. Firstly, the UD POS tagset is coarser-grained compared to the one in the PTB —the PTB has 48 syntactic categories, compared to the 17 categories in UD —and could miss important distinctions necessary for the task. Our first results, focused on English, suggest that coarser-grained syntactic annotation is sufficient, and can in fact lead to performance improvements on the task. Future work could include demonstrating the same on other languages such as French and Chinese, for which moderate-sized PDTB-style corpora have been annotated (Danlos *et al.*, 2015; Zhou & Xue, 2015).

1.1 Contribution

In the last three years, approaches to discourse parsing using manually engineered and selected features like those in Pitler & Nenkova (2009) and Lin *et al.* (2014) have taken a backseat to neural approaches using word embeddings. While our work draws upon these manually built features, we

1. For instance, the connective 'afterward', which denote a precedence relation between the units of text it joins, can be found lexicalized as 'afterwards', 'shortly afterward', and 'shortly afterwards' within the PDTB corpora.

believe it contributes towards the understanding of the role of syntactic information, specifically varying levels of tagset granularity, in the task of automatic discourse connective identification. We believe that this understanding can inform choices relating to the data processing pipeline and neural network architecture for a discourse parser.

To this end, our contributions are three-fold. Firstly, we demonstrate that it is possible for coarser-grained UD syntactic parses to perform as well as finer-grained PTB-style syntactic parses when used on the task. Secondly, we provide near-complete (see Section 5) replications, from the bottom-up, of the experiments conducted by Pitler & Nenkova (2009); Lin *et al.* (2014); Li *et al.* (2016) involving the automatic detection of discourse connectives. Thirdly and finally, in the same vein as Johannsen & Søgaard (2013); Lin *et al.* (2014); Braud *et al.* (2017) –who used both gold standard syntactic information as well as automatically-parsed information in their experiments to demonstrate their discourse parsers' performance in 'production' settings - our experimental set-up covers both gold-standard parses as well as predicted parses. This allows us to extend our analysis to discourse parsing under realistic settings. Although, in the absence of gold UD parses for sentences in the PDTB, we are only able to obtain approximations of such gold UD parses (see Section 4).

2 Related work

The PDTB consists of one million words contained in 40,600 articles obtained from the Wall Street Journal (WSJ) (Prasad *et al.*, 2008). The annotation approach in the PDTB focuses on identifying the local elements making up a coherent text. It identifies relations between two adjacent discourse units, which are linked by a relation that may be marked by a discourse connective. Approaches based on the PDTB and similar corpora, which focus on identifying local units of coherence in a text are referred to as shallow discourse parsing (SDP). This is in contrast with 'deep' approaches, using corpora such as the Rhetorical Structure Theory (RST) Discourse Treebank (Carlson *et al.*, 2001), which seeks to identify relations between discourse units extending across an entire document as well as hierarchically between groups of discourse units in the form of structured trees.

A hundred connective types and their lexicalization variants are annotated throughout the PDTB. These connective types fall in three syntactic classes, namely : subordinating and coordinating conjunctions such as *because* and *when* ; as well as *and*, as well as *or* respectively, and discourse adverbials such as *for example* and *instead*. On top of these, the spans of each connective's arguments, as well as the relation between them are also annotated. The relations marked by connectives fall within four broad classes (termed as 'senses') —Temporal, Contingency, Expansion, and Comparison, which are further categorized into finer types and sub-types (Prasad *et al.*, 2008). A PDTB parser typically addresses the identification of connectives first and it is only after this that the classification of the connectives' relations, and the spans of text they cover, are sequentially handled. Such modular approaches broadly adhere to the instructions in the annotation manual [2] used in the annotation process for the PDTB version 2.0 (Polakova *et al.*, 2017). Importantly, this is possible due to the local coherence approach taken by the PDTB, which limits but does not preclude the direct application of an automatic connective identification module on parsers for other corpora that focus on more global levels of coherence [3].

2. https://www.seas.upenn.edu/~pdtb/PDTBAPI/pdtb-annotation-manual.pdf

3. for e.g. RST-based corpora, where until recently, it was not seen as necessary to annotate connectives within the RST corpora. With the release of the RST Signaling Corpus (RST-SC) in 2015, the annotation of connectives and other information that signal a coherence relation in a text were included and could be used by researchers in the parsing of the RST corpora.

The task of connective identification attracted attention when the PDTB was first released. Pitler & Nenkova (2009) proposed the use of simple lexical and morpho-syntactic [4] features in a binary classifier. They showed that simply using the connective string already leads to high performance (85.86% in accuracy), meaning that many forms are very often in discourse use. Adding morpho-syntactic information leads to a very high accuracy of 96.26%. This led to assumptions that the task was solved. However, later work, especially (Lin *et al.*, 2014) and (Johannsen & Søgaard, 2013), demonstrate that the task was in fact easy mainly for (1) a few connectives that occur highly frequent and (2) within a fully gold setting ; when considering a realistic setting with predicted (PTB) syntactic information, a performance drop of 4.5 percentage points in macro F1 was observed. They also observed that connective classifiers typically struggle to predict accurately connective strings that occur less frequently ; for instance, Lin *et al.* (2014)'s classifier could only reach a 43.2% F1 score for the connective 'ultimately', which is in the fiftieth percentile amongst the 100 connectives in terms of frequency (Johannsen & Søgaard, 2013).

Lin *et al.* (2014) noted that, accordingly, *"high performance* [in the identification of discourse connectives] *is crucial to mitigate the effect of cascaded errors downstream"*. They found that accumulated errors (i.e. without replacing predictions from the previous module with gold standard information drawn from the corpora) in their PDTB parser pipeline led to a drop in the F1 score of the pipeline's last module, the attribute span labeler, from 79.68% to 72.27% (a 7.41 percentage point drop). In addition, we note that although the PDTB is the largest-sized discourse corpora currently available, it is comprised of texts drawn from the financial news domain, and connectives that are infrequent in the PDTB could become frequent in another domain outside of the PDTB.

Pitler & Nenkova (2009)'s work was extended by Lin *et al.* (2014), who obtained improved results by adding more lexical and syntactic features (such as the strings and POS tags of the connective's neighbours). They also included information about a sentence's or clause's structural properties [5]. In the most recent shared task focused on the PDTB, Li *et al.* (2016) reported obtaining higher F1 scores compared to Lin *et al.* (2014) (see Table 2) and, to our knowledge, the highest published F1 scores for the task of PDTB explicit discourse connective identification. They used a similar, but smaller, set of features as Lin *et al.* (2014), which left out features relating to sentence/clause structure.

Other methods cast the connective identification task as a sequence labeling problem, making use of methods such as conditional random field models ; although to our knowledge, the results have not reached the level obtained with binary classification approaches and have not gained traction. For instance, Stepanov & Riccardi (2016)'s CRF system only obtained an F1 score of 92.43% on the PDTB test set ; compared to the 98.92% in Lin *et al.* (2014)'s binary classification system. Additionally, neural network approaches, leveraging word vectors, have also taken hold in recent years (Xue *et al.*, 2015). With regards the latter, efforts have also been made to produce an end-to-end approach to parsing the PDTB dataset (covering connective identification as well as 'downstream' discourse parsing tasks such as argument identification and sense labeling) (Weiss & Bajec, 2018).

4. The features used by Pitler & Nenkova (2009) are : (a) `Self Category` : the syntactic category of the highest node on the parse tree that covers only the connective phrase ; (b) `Parent Category` : the parent node of the self category ; (c) `Left Sibling Category` : the syntactic category immediately to the left of the self category ; (d) `Right Sibling Category` : the syntactic category immediately to the right of the self category. (e) `Right Sibling Contains a VP` (verb phrase) ; and (f) `Right Sibling Contains a Trace`, as well as the interaction between these features.

5. The features (Lin *et al.*, 2014) added were : (a) the connective POS tag (`CPOS`) ; (b) the token before the connective string (`Prev1`) + the connective string (`C-string`) ; (c) the POS tag of Prev1 (`Prev1POS`) ; (d) `Prev1POS` + `CPOS` ; (e) `C-string` + the token following C-string (`Next1`) ; (f) the POS tag of Next1 (`Next1POS`) ; (g) `CPOS` + `next1POS` ; (h) the `path of the connective's parent to the root` ; (i) the `compressed path of the connective's parent to the root`. '+' indicates interaction between features.

Regarding studies of the impact of syntactic information in discourse parsing, Braud *et al.* (2017) analyzed the role of syntax in the discourse parsing tasks. They found that using UD syntactic information led to a loss in performance compared to when gold standard PTB syntactic information is used; albeit this was for a different task of sentence boundary identification on a different discourse corpora, the RST Discourse Treebank (RST-DT). They observed that PTB POS tags including 'WDT' (wh-determiner) and 'WP$' (possessive wh-pronoun) collapse into one single POS tag, 'DET' (determiner) in UD, and the decreased granularity led to an ambiguous signal that is is a source of increased error when using UD information for their task.

Finally, recent work on learning representations for sentences have found relevance in using explicit discourse connectives identification as a task to guide the learning of such representations. Nie *et al.* (2019) and Sileo *et al.* (2019) trained sentence encoders using large corpora of sentence pairs with discourse markers between them and achieved state-of-the-art results on their approaches. The former worked with a corpora containing 15 of the most frequently occurring discourse connectives, whereas the latter relied on a heuristic to identify discourse connectives candidates, of which some of them have not been annotated in any dataset such as the PDTB.

3 Approach

Pitler & Nenkova (2009) approached the automatic connective identification task by training a binary classifier using a set of features that includes, and combines, lexical and syntactic information of the connective candidate. They observed that discourse connectives occur in *"specific syntactical contexts"*; many connectives take a subordinate clause as one of its arguments (for example, in the sentence "After I went to the store, I went home") and the PTB POS tag 'SBAR' marks such subordinate clauses. It was observed that such syntactic information are indicators of a connective candidate being in discourse usage. However, a single POS tag alone may not be sufficient to disambiguate between whether a connective candidate is in discourse or non-discourse usage [6], and Pitler & Nenkova (2009) found that extending the set of features to include syntactic information from a wider context around a connective candidate improves the performance of a classifier.

Their work have become seminal for the task and is cited by subsequent researchers working on connective identification. To study whether UD information is sufficient for the task, we adopt their general approach as the basis for our experiments. We also include the work of Lin *et al.* (2014) whose added syntactic features (see Section 2) meaningfully improved on the performance of Pitler & Nenkova (2009), as well as Li *et al.* (2016), who reported the highest F1 score on the task with the PDTB test set during the CoNLL 2016 Shared Task (Xue *et al.*, 2016).

In practical terms, we reproduced the pre-processing and feature engineering pipelines of these three authors as well as obtained approximately gold-standard UD information for the PDTB, which is not available. This allowed us to (1) validate our reconstruction of their connective identification pipelines, (2) isolate the impact of differences in our classifier and hyperparameter settings with these authors', and (3) have a broad-based set of experimental set-ups to study the impact of using UD

6. For example, the two words 'instead' as well as 'and' are discourse connective candidates in the sentence "NASA won't attempt a rescue; **instead**, it will try to predict whether any of the rubble will smash to the ground and where." (Pitler & Nenkova, 2009), though only the former is being used as a discourse connective. This is despite the syntactic category (the POS tag immediately encapsulating a connective candidate) for 'and' being 'SBAR' too; and is cited by them as demonstrating that syntactic information from a wider context is necessary for connective disambiguation.

instead of PTB information. We describe each part of our approach in the next sections.

3.1 Features and data representation

For our experiments, we sought to reproduce our settings to be as similar as those in (Pitler & Nenkova, 2009; Lin *et al.*, 2014; Li *et al.*, 2016), based on the information from their published papers as well as code available online [7]. We used the same data representation as these authors; their experiments utilized a feature-based one-hot encoded approach where each data point is represented by a vector of a fixed size, which corresponds to the number of one-hot features obtained from the training set. The presence of a particular feature in a data point is marked by a value of one in its position on the vector, and zero otherwise. This results in a sparse representation of the data point. The numbers of features present in each of the three experiments mentioned in this section are listed in Table 1.

To the extent possible, we also used the feature sets they used in their experiments (see footnote 4 and footnote 5 for the list of the features they used and a description of them). However, they did use a number of PTB-related features for which there are no direct equivalents in UD. For instance, one feature used by both of Pitler & Nenkova (2009) and Lin *et al.* (2014), is built from the syntactic category that the connective string is constituent of [8]. There is no corresponding category in the UD dependency grammar approach. Similarly, Lin *et al.* (2014) include two other features built from the collection of syntactic categories in the path between the connective and the root of the sentence [9].

As such, we conducted two groups of experiments (see Section 5) to be able to study in isolation the impact of switching between PTB and UD information. One of the group (see Section 5.2) involves the use of UD information and because there are no direct equivalents in UD of the PTB-style features in Pitler & Nenkova (2009)'s and Lin *et al.* (2014)'s experiments, we did not include their feature sets in our second group of experiments. Instead, we used Li *et al.* (2016)'s feature set there, although with two sets of modifications. The first modification is to exclude the feature relating to the parent constituent of the connective candidate [10]. The second modification replaces the remaining features with UD information.

In summary, after taking these into consideration, our second group of experiments were conducted with the maximal set of features that are present in Li *et al.* (2016)'s PTB-style features as well as where comparable information can be obtained from UD dependency-based parses. In addition, we also conducted each group of experiments with gold and automatically parsed information alternately, to study the effect of the connective classifier in 'production' settings.

3.2 PTB to UD conversion

UD (Nivre *et al.*, 2016) is a syntactic framework introduced in 2016 [11]. The UD project seeks to establish a framework that allows a consistent syntactic annotation approach across languages around the world, while having the flexibility and capabilities to capture linguistic phenomena in these languages. As of November 2019, there are 157 UD treebanks in 90 languages [12]. Besides the

7. https://github.com/linziheng/pdtb-parser
8. 'Parent Category', see footnote 4
9. See points (h) and (i) in footnote 5
10. 'Self Category', see footnote 4
11. Although it traces its roots to the Stanford Dependencies framework that was released in 2008.
12. https://universaldependencies.org/

Experiment		Number of dimensions					
		PTB Gold1	PTB Auto1	PTB Gold2[1]	PTB Auto2[1]	UD Gold	UD Auto
P & N 2009	C*	101	-	-	-	-	-
	CSynI**	1,787	554	-	-	-	-
Lin et al 2014		66,975	51,584	-	-	-	-
Li et al 2016		33,308	32,971	33,216	32,945	32,015	32,050

* Connective string only.

** Connective string, syntactic features and interaction between features.

1 This excludes the Self Category feature which relates to the parent constituent of the connective candidate.

TABLE 1 – Number of dimensions in the feature sets used for each of the experiments.

difference in granularity of their POS tagsets, the manner that UD and PTB frameworks capture information about the syntactic relations between words is different; the former adopts a dependency grammar approach whereas the PTB hews to a constituency grammar approach. As a result, certain features used in PTB-based approaches to the task may not be obtainable from UD information.

The WSJ articles that make up the PDTB are the same as those in the PTB. Accordingly, gold standard PTB parses are available and we used these for the parts of our feature extraction processes with PTB-based features. However, gold standard, manually-annotated, UD parses for the sentences in the PDTB are not available and we had to approximate these. Although the organizers of the 2015 CoNLL Shared Task on discourse parsing with the PDTB provided dependency grammar-style syntactic parses, these are of the Stanford Dependencies framework and understood to be automatically parsed (instead of manually annotated). Using these would mean that we would not be able to model current use-in-production settings.

As such, to approximate gold standard UD parses, we used an earlier version of the Stanford CoreNLP package with an option that is intended to convert PTB parses to UD [13]. We obtained (1) gold UD version 1.0 parses [14] using the gold PTB parses, as well as (2) separately, automatically generated parses using the UniversalDependenciesConverter in the Stanford CoreNLP package.

We note that these UD parses obtained are 'approximate' as errors have been observed in conversions from PTB to UD; although the conversion of most POS tags in PTB to UD is *"almost trivial"* (Peng & Zeldes, 2018), there are errors which mainly relate to the conversion of constituent categories to dependencies relations. There are certain words with PTB POS tags that are not possible to map to UD POS tags without additional UD dependency information. For instance, determiners are tagged as a 'DT' in the PTB framework, but may be tagged as 'DET' or 'PRON' in UD depending on whether the determiner word is used independently or not, and this requires dependency relation information during the conversion process. Peng & Zeldes (2018) note that conversion from PTB to UD dependency relations sees also errors increase on out-of-domain input, *"in all genres, including when using gold constituent trees, primarily due to underspecification of phrasal grammatical functions"* .

13. https://nlp.stanford.edu/software/stanford-dependencies.shtml

14. We were not able to obtain parses with more recent versions of UD (i.e. UD2.0 and later) as there are no converters available currently to convert between PTB and UD 2.0 and later versions. However, the applicability of our experimental set-up on UD 2.0 data is not affected; our work isolates the impact of changing PTB POS tag for UD 1.0 POS tag, and the changes in the POS tagset from UD 1.0 to UD 2.0 relates to four specific tags - AUX, PRON, DET and PART, which are not in themselves signal for connectives. We also verified that none of our UD featuresets contain features with one of these four POS tags affected by changes in UD 2.0.

4 Settings

In this section we outline the settings of our experiments. Our experiments were conducted with version 2.0 of the PDTB, which were made available by organisers of the CoNLL 2015 Shared Task [15]. We kept to the train-development-test split that were prescribed by the PDTB creators [16]. Pitler & Nenkova (2009); Lin *et al.* (2014) and Li *et al.* (2016) used implementations of MaxEnt classifiers in two NLP machine learning packages [17]. Both of these are Java-based machine learning packages with specific requirements on the format of the training data. We chose instead to implement the feature extraction and classification pipeline in Python, using a popular machine learning package scikit-learn (Pedregosa *et al.*, 2011). We used the latter's LogisticRegressionCV classifier under a multinomial setting which makes it equivalent to a MaxEnt classifier [18]. To allow the results between each set of experiments to be as directly comparable as possible, we did not implement any hyperparameter optimization procedures.

5 Experiments

Our experiment is composed of two sub-groups. The first group is a complete reproduction of Pitler & Nenkova (2009); Lin *et al.* (2014); Li *et al.* (2016)'s pre-processing and classifier pipelines with the entire feature set that each of them used. We describe and discuss these in the following sections.

5.1 Replication experiments

This first group relates to our own reproduction of the experiments described in the works of Pitler & Nenkova (2009), Lin *et al.* (2014), and Li *et al.* (2016). We do this so as to : (1) validate our data pre-processing and feature extraction pipeline by checking that our subsequent results are within the range of the established standards for the task ; and (2) produce results controlling for our use of a different machine learning package and parameter settings (see Section 4).

The results we obtained (see Table 2) indicate that our reproduction of the pro-processing and feature engineering pipelines are in line with the original authors'. In fact, we obtained almost across-the-board better results compared to the original authors ; in one case, by 3.83 percentage points. We believe that these improved results likely stem from minor variations between our experimental set-up and theirs ; for example, it could be due to the : (1) choice of the MaxEntLogistic Regression implementation, (2) hyperparameter settings such as the specification of class weights, and/or (3) choice of the type of F1 score reported on.

However, Li *et al.* (2016) stated that their F1 result of 98.92% on the test set was *"according to official evaluation* [by the shared task organizers]", but our reproduction returned a lower F1 score of

15. https://www.cs.brandeis.edu/~clp/conll15st/index.html
16. Sections 2 to 21 are used for training, with sections 22 and 23 used for development and testing respectively
17. Pitler & Nenkova (2009) used the Mallet package http://mallet.cs.umass.edu/, whereas Lin *et al.* (2014) and Li *et al.* (2016) used an implementation by OpenNLP https://opennlp.apache.org.
18. MaxEnt models learn parameters that, as their suggests, maximizes the entropy of the classes within the data. They have been shown to be equivalent to multinomial logistic regression approaches (Manning & Klein, 2003). We also kept most of the default values specified in scikit-learn for the rest of the settings. The settings that we changed from the default values include specifying : (1) that ten-fold cross validation with the data, which the original authors also conducted during their training steps ; and (2) the class weight, which is the distribution between the negative and positive examples in the training set.

Data split	Set-up	Feature set		
		P & N 2009	Lin et al 2014	Li et al 2016
Train	Author's	94.19%	95.36%	*
	Ours	95.28%	99.19%	97.88%
Test	Author's	*	*	98.92%
	Ours	95.10%	97.22%	92.52%

* Result not published.

TABLE 2 – F1 score reported by authors and obtained by our replication of their experiments. Our scores are weighted F1. None of the authors mentioned if their scores were computed as weighted, micro or macro F1.

92.52%. A summary article of the shared task by its organizers (Xue *et al.*, 2016) lists their system as having a performance of 94.71% F1 score on the test set. We were unable to identify any further information regarding these differences, but note that our experimental set-up (see Section 3) allows us to isolate the impact of such differences when studying the effect of using UD instead of PTB information, as well as replacing gold-standard information with automatically parsed information.

5.2 UD vs PTB information

The second group in our experiment involves the set-up used in Li *et al.* (2016), which is the most recent of the three works. Here, we removed one feature, 'Self Category' (see footnote 4) from the set-up in order to ensure a comparability between features built with PTB and UD syntactic information. In this group, we built Li *et al.* (2016)'s PTB-based features with UD instead. Additionally, we conducted this set of experiment with features built from gold as well as automatically produced parses for both the PTB and UD experiments.

We found that there was no performance loss in switching from the use of PTB to UD POS tags in the task of discourse connective identification on the PDTB. In fact, we found a minor gain in F1 scores of 1.8% point on the PDTB test set when switching from gold PTB to gold UD information. We found a very minor decrease (a 0.25 percentage point drop on the test set) in moving from gold to automatically parsed UD information, which is expected.

Surprisingly, we found the move from gold PTB information to automatically parsed UD information brought about a 1.55 percentage point improvement in the weighted F1 score on the test set. Nonetheless, we have reason to believe that these changes in results are statistically significant. We conducted Wilcoxon signed rank-tests between the outputs of these models and they returned p-levels of below 0.05, which is sufficient to reject the null hypothesis that the outputs are similarly distributed. These changes in the results, from using UD instead of PTB information, are presented in Table 3, whereas a fuller report of the weighted F1, the accuracy as well as macro and micro F1 scores using the features in Li *et al.* (2016) can be found in Table 4.

5.3 Discussion

Our findings shows that a logistic classifier does not require the fine level of granularity present in the PTB in order to disambiguate whether a connective candidate is in discourse usage or not. In particular,

Parameter	Change in weighted F1 score, % points	
	Train	Test
PTB Gold to UD Gold*	0.26% (97.85% to 98.11%) $p \ll 0.001$	1.8% | (92.21% to 94.01%) $p \ll 0.001$
UD Gold* to UD Auto	-0.08% | (98.11% to 98.03%) $p < 0.001$	-0.25% | (94.01% to 93.76%) $p < 0.04$
PTB Gold to UD Auto	0.18% | (97.85% to 98.03%) $p \ll 0.001$	1.55% | (92.21% to 93.76%) $p \ll 0.001$

TABLE 3 – Changes in weighted F1 scores, between syntactic framework choices, on the feature set used in Li *et al.* (2016). The p-values reported in the table are results of Wilcoxon signed rank-tests between the outputs of each model pair. *UD Gold above refers to the use of data approximated automatically from PTB Gold data.

we note that the move to using UD resulted in a reduction of about 1,200 features. The number of features fell from an initial 32,015 when using PTB syntactic information, to 33,216 features or about a 3.5% reduction in the feature set size when moving to the use of UD syntactic information. The granularity in the PTB POS tags set leads to an increased number of features compared to when UD information is used. It appears to us that, this in turn led to a more complex decision boundary when using PTB information, which the classifier found harder to learn.

To examine this further, we carried out a per-connective error analysis on our results on the test set. Figure 1 shows the distribution of the wrongly classified connectives when predicted with gold PTB parses compared with when gold UD parses are used. We observe that the classifier remains confused by the same connectives in both cases, but that the 1.8 percentage point improvement in weighted F1 score is due to an increase in correct predictions that are more or less evenly distributed across the connectives [19]. This lends support to our hypothesis that the reduction in features are helping the classifier to better model the decision boundary for the connectives. Likewise, as shown in Figure 2, we observe a similar distribution of prediction errors when comparing between the use of gold PTB and automatically parsed UD information.

We note however, that these results were based on experiments conducted on the features used in Li *et al.* (2016) which exclude certain PTB-style structural information (e.g. connective to root) used in Lin *et al.* (2014). The effect of this is a loss of 5.41 % points in the F1 weighted score [20] when moving from the training set to the test set. In comparison, for Lin *et al.* (2014), this loss is only 1.99 % points (98.55% to 96.56%) when predicting on the test set. This suggests that the Lin *et al.* (2014)'s feature set produces a more robust connective identifier that generalizes better for unseen data.

While a direct replacement of such structural information is not available in UD, we note that some success have been observed in using syntactic structural information in UD ('supertags' which capture information such as incoming and outgoing dependency relations for a word), for the task of sentence segmentation in discourse parsing (Braud *et al.*, 2017), and that this could be an area of future research to extend our findings here.

19. The table shows that about half of the increase in correct predictions are for the word 'but' ; however the reduction remains proportional across all the connectives as 'but' is the most frequently present of the connective strings in the PDTB, and of its occurrences, more than 70% are as a discourse connective (Johannsen & Søgaard, 2013).

20. From 97.88% on the training set to 92.52% the test set. This is using PTB gold parses. The same figures for PTB Auto are : 97.83% (train) and 92.42% (test).

Experiments		Accuracy	F1-macro	F1-micro	F1-weighted
PTB Gold	train	98.87%	97.36%	97.87%	97.85%
	test	92.37%	90.72%	92.37%	92.21%
PTB Auto	train	97.84%	97.33%	97.84%	97.83%
	test	92.54%	90.89%	92.54%	92.37%
UD Gold	train	98.12%	97.69%	98.12%	98.11%
	test	94.04%	92.94%	94.04%	94.01%
UD Auto	train	98.04%	97.58%	98.04%	98.03%
	test	93.81%	92.63%	93.81%	93.76%

TABLE 4 – Accuracy, F1 (macro, micro and weighted) scores with features from Li *et al.* (2016).

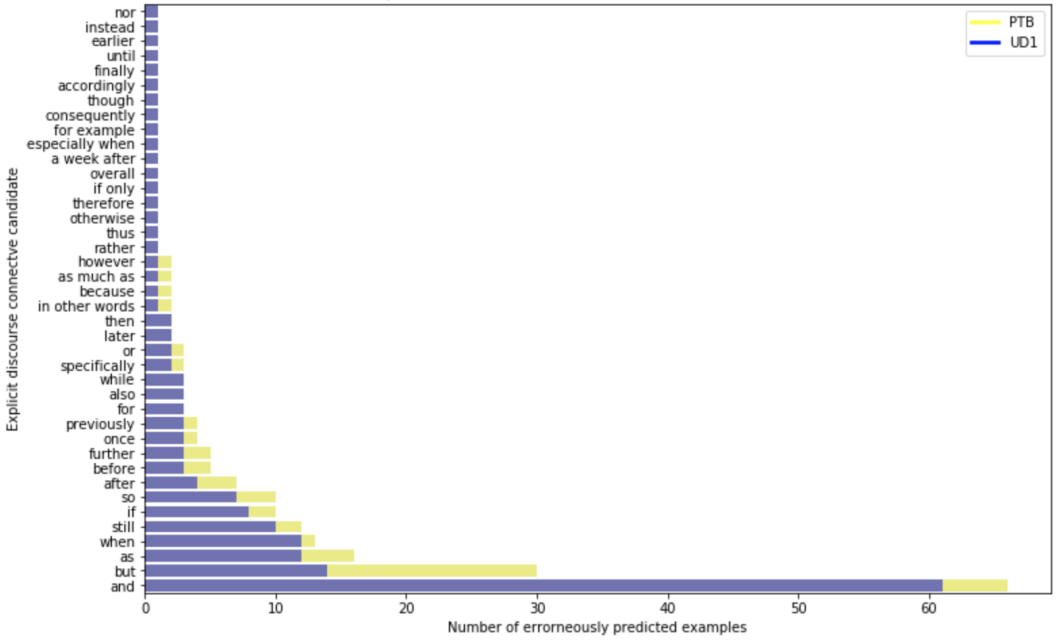


FIGURE 1 – Connective error count for PDTB test set, using `gold PTB` (yellow) and `gold UD1` (blue) syntactic information and the feature set used in Li *et al.* (2016), without the 'Self-Category' feature.

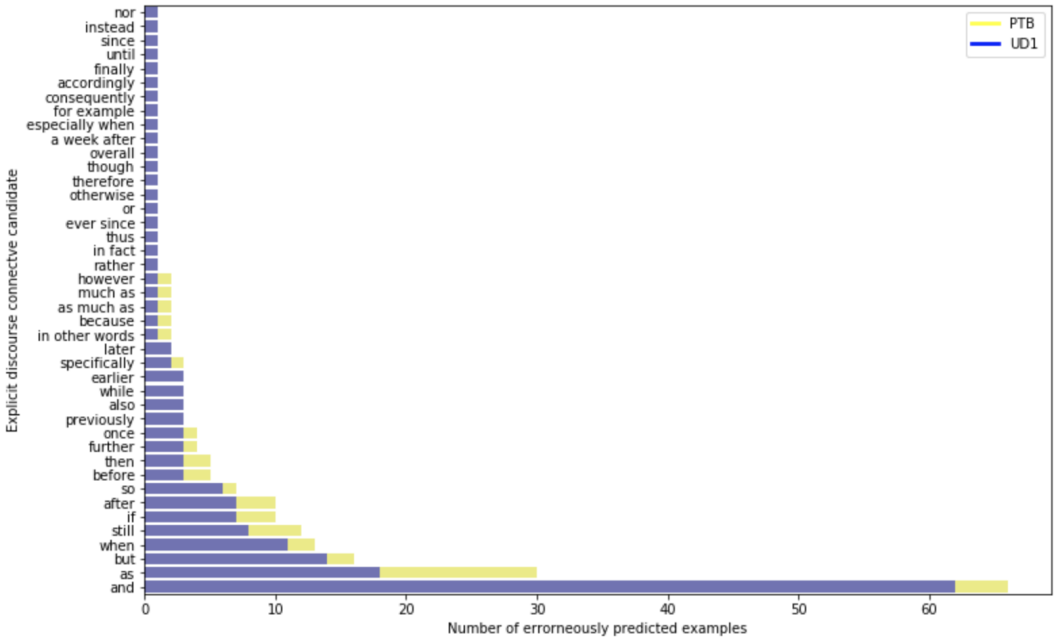


FIGURE 2 – Connective error count for PDTB test set, using `gold PTB` (yellow) and `automatically parsed UD1` (blue) syntactic information and the feature set used in Li *et al.* (2016).

6 Conclusion

We reproduced the experiments of Pitler & Nenkova (2009), Lin *et al.* (2014) and Li *et al.* (2016) in order to study the impact of using UD instead of PTB syntactic information. Our results, under a binary classification setting using a logistic regression classifier, show that UD syntactic information is a viable alternative to PTB information. Our analysis indicate that the improvement is likely because it is easier for the classifier to model the decision boundary when using coarser-grained UD syntactic information, as it leads to a reduction in the number of features needed to represent the data. Our code for the connective classifier can be found at : `https://gitlab.inria.fr/andiamo/marta-v2`.

Acknowledgements

We thank Chloé Braud for her patient guidance, inspiration and nurturing encouragement throughout our undertaking of this work, as well as for her kind help in reviewing the manuscript for this article and the invaluable suggestions she shared with us in the process. We also thank the two anonymous reviewers for their helpful comments.

Références

BRAUD C., LACROIX O. & SØGAARD A. (2017). Does syntax help discourse segmentation ? not so much. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language*

Processing, p. 2432–2442, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/D17-1258.

CARLSON L., MARCU D. & OKUROVSKY M. E. (2001). Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*. DOI : https ://doi.org/10.3115/1118078.1118083.

DANLOS L., COLINET M. & STEINLIN J. (2015). Fdtb1, première étape du projet « french discourse treebank » : repérage des connecteurs de discours en corpus. *Discours*, **17**. DOI : 10.4000/discours.9065.

JOHANNSEN A. & SØGAARD A. (2013). Disambiguating explicit discourse connectives without oracles. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, p. 997–1001, Nagoya, Japan : Asian Federation of Natural Language Processing.

LI Z., ZHAO H., PANG C., WANG L. & WANG H. (2016). A constituent syntactic parse tree based discourse parser. In *Proceedings of the CoNLL-16 shared task*, p. 60–64, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/K16-2008.

LIN Z., NG H. T. & KAN M.-Y. (2014). A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, **20**(2), 151–184. DOI : 10.1017/S1351324912000307.

MANNING C. & KLEIN D. (2003). Optimization, maxent models, and conditional estimation without magic. USA : Association for Computational Linguistics.

MARCUS M. P., SANTORINI B. & MARCINKIEWICZ M. A. (1993). Building a large annotated corpus of English : The Penn Treebank. *Computational Linguistics*, **19**(2), 313–330.

NIE A., BENNETT E. & GOODMAN N. (2019). DisSent : Learning sentence representations from explicit discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4497–4510, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1442.

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

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M. & DUCHESNAY E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

PENG S. & ZELDES A. (2018). All roads lead to UD : Converting Stanford and Penn parses to English universal dependencies with multilayer annotations. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, p. 167–177, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

PITLER E. & NENKOVA A. (2009). Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, p. 13–16, Suntec, Singapore : Association for Computational Linguistics.

POLAKOVA L., MÍROVSKÝ J. & SYNKOVÁ P. (2017). Signalling implicit relations : A pdtb - rst comparison. *Dialogue and Discourse*, **8**. DOI : 10.5087/dad.2017.210.

PRASAD R., DINESH N., LEE A., MILTSAKAKI E., ROBALDO L., JOSHI A. & WEBBER B. (2008). The Penn discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference*

on Language Resources and Evaluation (LREC'08), Marrakech, Morocco : European Language Resources Association (ELRA).

SILEO D., VAN DE CRUYS T., PRADEL C. & MULLER P. (2019). Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 3477–3486, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1351.

STEPANOV E. & RICCARDI G. (2016). UniTN end-to-end discourse parser for CoNLL 2016 shared task. In *Proceedings of the CoNLL-16 shared task*, p. 85–91, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/K16-2012.

WEISS G. & BAJEC M. (2018). Sense classification of shallow discourse relations with focused rnns. In *PloS one*. DOI : https ://doi.org/10.1371/journal.pone.0206057.

XUE N., NG H. T., PRADHAN S., PRASAD R., BRYANT C. & RUTHERFORD A. (2015). The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, p. 1–16, Beijing, China : Association for Computational Linguistics. DOI : 10.18653/v1/K15-2001.

XUE N., NG H. T., PRADHAN S., RUTHERFORD A., WEBBER B., WANG C. & WANG H. (2016). CoNLL 2016 shared task on multilingual shallow discourse parsing. In *Proceedings of the CoNLL-16 shared task*, p. 1–19, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/K16-2001.

ZHOU Y. & XUE N. (2015). The chinese discourse treebank : a chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, **49**. DOI : 10.1007/s10579-014-9290-3.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Transformations syntaxiques entre niveaux de simplification dans le corpus Newsela

Rita Hijazi[1, 2]
(1) LPL, 13090 Aix-en-Provence, France
(2) LIS, 13397 Marseille, France
rita.hijazi@etu.univ-amu.fr

RÉSUMÉ

La simplification de textes est une tâche complexe du traitement automatique des langues. Depuis quelques années, des corpus parallèles de textes originaux et simplifiés sont proposés, permettant d'apprendre différents types d'opérations de simplification à partir de corpus. Dans le but de pouvoir développer et évaluer des systèmes de simplification automatique de textes, cet article s'intéresse au corpus Newsela, un corpus parallèle de textes en langue anglaise avec quatre niveaux de simplification. Nous présentons en détail ce corpus et étudions les différentes transformations caractérisant le passage d'un niveau de simplification à l'autre sur un sous-ensemble de textes, en nous intéressant plus particulièrement aux transformations syntaxiques.

ABSTRACT

Syntactic transformations between simplification levels in the Newsela corpus.

Text simplification is a complex task of Natural Language Processing. Research into this topic has many potential practical applications. For several years now, parallel corpora of complex–simple paired sentences have been developed to provide examples of structural transformations and particularly suitable for text simplification. To develop and evaluate ATS systems, this article focuses on the Newsela corpus, a parallel corpus of English texts with four levels of simplification. We present this corpus in detail, and we study the different transformations characterizing the passage from one level of simplification to another on a subset of texts. We specifically focus on syntactic transformations.

MOTS-CLÉS : Corpus parallèle, simplification de textes, analyse du corpus.
KEYWORDS: Parallel corpora, text simplification, corpus analysis.

1 Introduction

La simplification de textes (SAT) est un domaine du traitement automatique des langues (TAL) qui suscite l'intérêt dans la communauté depuis quelques années. L'objectif est de rendre des textes plus abordables tout en garantissant l'intégrité sémantique de leur contenu. Saggion (2017) définit la SAT comme étant le processus de transformation d'un texte en un autre texte qui véhicule le même contenu sémantique, afin de le rendre plus facile à lire et à comprendre par un public cible.

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 137–150.
HAL : hal-02786194.

La SAT peut être adressée à des lecteurs humains faisant face à différents types de difficultés de lecture, par exemple, les apprenants de langues (Petersen et Ostendorf, 2007 ; Burstein, 2009), les personnes souffrant d'aphasie (Devlin and Tait, 1998 ; Carroll et al., 1998), de dyslexie (Rello et al., 2013) ou d'autisme (Evans et al., 2014). La SAT peut aussi être utilisée comme étape de prétraitement pour d'autres tâches de TAL, telles que l'analyse syntaxique (Chandrasekar et al., 1996), le résumé automatique (Vanderwende et al., 2007 ; Silveira et Branco, 2012) et la traduction automatique (Hasler et al., 2017).

Un des problèmes majeurs pour la construction de systèmes de SAT efficaces est l'absence de corpus annotés en niveaux de difficulté ou, tout au moins, des versions parallèles originales et simplifiées. En effet, une étape préalable et importante vers la construction des systèmes de SAT est l'analyse et la comparaison de versions parallèles de textes simplifiés originaux, afin d'examiner quels types de changements doivent être appliqués et pour quel public et quelles ressources sont nécessaires pour les mettre en place automatiquement. Nous nous intéressons ici au corpus Newsela (Newsela, 2016) dédié à cette tâche. Il s'agit d'un corpus parallèle de textes en anglais avec quatre niveaux de simplification. Nous présentons en détail ce corpus et étudions les différentes transformations caractérisant le passage d'un niveau de simplification à l'autre sur un sous-ensemble de textes, en nous intéressant plus particulièrement aux transformations syntaxiques.

Cet article est structuré de la façon suivante. Dans la section 2, nous définissons le contexte du travail et nous présentons en détail le corpus Newsela, un corpus parallèle dédié à la simplification de textes en anglais avec quatre niveaux de simplification. Dans la section 3 nous présentons la méthodologie que nous avons retenue pour l'étude des différents niveaux de simplification de ce corpus. Dans la section 4 nous proposons une analyse des opérations de simplification mises en œuvre pour passer d'un niveau de simplification à l'autre dans le corpus Newsela. Nous concluons en présentant plusieurs perspectives à ce travail.

2 Contexte

2.1 Des corpus pour la simplification des textes

Force est de constater que les corpus parallèles version originale *vs* version simplifiée ne sont pas nombreux, bien que pour l'anglais, des initiatives différentes ont vu le jour dans différents domaines (pas forcément en TAL). L'initiative Plain English[1], par exemple, est née dans les années 1970 pour faciliter la compréhension des textes officiels administratifs. Il s'agit de directives qui peuvent en principe s'appliquer à toutes les langues, par exemple : garder le sujet, le verbe et l'objet ensemble ; expliquer une seule idée par phrase ; utiliser des phrases courtes ; utiliser la voix active ; etc.
En TAL, des ressources existent pour le français, comme Vikidia[2] (Wikipédia Junior) qui a été utilisée par Brouwers et ses collaborateurs (2012) afin d'établir une typologie des règles de simplification. Vikidia est un corpus destiné aux jeunes de huit à treize ans et rassemble des articles plus accessibles, tant au niveau de la langue que du contenu (Brouwers et al., 2012). Ce corpus comprend à ce jour plus de 29 900 articles.
Une autre ressource existante est le corpus Alector[3] (Gala et al., 2020a). Il s'agit d'une collection de 79 textes littéraires (contes, histoires) et scientifiques (documentaires) originaux ainsi que leurs

1 https://www.plainlanguage.gov/
2 http://fr.vikidia.org/
3 https://corpusalector.huma-num.fr/

équivalents simplifiés. Les textes ont été choisis parmi une variété de supports disponibles pour les élèves des écoles primaires françaises, particulièrement les apprenants du cours élémentaire 1 et 2 (CE1 et CE2), et cours moyen 1 (CM1). Les 79 textes originaux ont tous subi des simplifications aux niveaux lexical, morphologique, syntaxique et discursif. Les simplifications ont été faites manuellement par une équipe d'experts, les corpus ont été testés dans des écoles (plus de mille élèves) dans le but d'obtenir des résultats sur l'impact des simplifications sur la lecture et la compréhension[4].

Au niveau de la SAT, on distingue plusieurs étapes. L'une des sous-tâches est également la simplification lexicale (Specia et al., 2012), qui consiste à remplacer les mots par des synonymes plus simples. La simplification est parfois une forme de paraphrase dans laquelle une phrase est reformulée en une phrase linguistiquement plus simple tout en conservant le sens de la phrase d'origine. Les paraphrases pour la simplification sont généralement extraites à partir de corpus parallèles. En anglais, Creutz (2018) a proposé Opusparcus[5] (*OpenSUbtitleSPARaphraseCorpUS*), un corpus de paraphrases pour six langues européennes : l'allemand, l'anglais, le finnois, le français, le russe et le suédois. Les ensembles de données ont été extraits d'OpenSubtitles2016[6] (Lison et Tiedemann, 2016), qui est une collection de sous-titres traduits de films et d'émissions de télévision. Pour chaque langue, les données sont divisées en trois types d'ensembles de données apprentissage, développement et évaluation. Les données d'apprentissage sont composées de millions de paires de phrases, et ont été compilés automatiquement. Les ensembles de développement et d'évaluation sont constitués de paires de phrases qui ont été vérifiées manuellement ; chaque ensemble contient environ 1000 paires de phrases (Creutz, 2018). Ce corpus, du fait de son contenu (paraphrases) peut être utilisé à des fins de développement ou de test d'un système de SAT.

La paraphrase a déjà été prise en compte dans des systèmes de SAT à des fins éducatives. Ces systèmes reposent souvent sur des règles de transformation définies par des experts. Inui et ses collaborateurs (2003) répondent aux besoins des apprenants sourds de l'anglais et du japonais écrits en paraphrasant des textes en supprimant les structures syntaxiques difficiles pour ce groupe d'apprenants (Inui et al., 2003). Le but du projet *Practical Simplification of English Text* (PSET) est de paraphraser les textes des journaux pour les personnes aphasiques (Canning et al., 2000). Max et ses collaborateurs (2006) ciblent les rédacteurs de textes pour les lecteurs souffrant de troubles du langage avec un système de simplification de texte interactif intégré dans un traitement de texte suggérant des simplifications tout en permettant à l'auteur du texte de garder le contrôle sur son contenu (Max, 2006). Le service de test pédagogique (*Educational Test Service* ETS) a développé l'outil d'adaptation automatique de texte (*Automatic Text Adaptation* ATA ; Burstein et al., 2007). Ce système ne simplifie pas directement le texte d'origine mais fournit plutôt une aide à la lecture via des adaptations de texte en anglais et/ou en espagnol qui sont affichées avec le texte original. Ces adaptations incluent la prise en charge du vocabulaire, les notes marginales et la synthèse vocale.

Une deuxième sous-tâche de SAT est la simplification syntaxique ayant pour but d'identifier et de transformer de longues phrases contenant des phénomènes syntaxiques qui peuvent nuire à la lisibilité pour certaines personnes en paraphrases plus simples qui ne contiennent pas ces phénomènes. La majorité des méthodes de simplification syntaxique proposées reposent sur un ensemble de règles de transformation définies manuellement pour être appliquées aux phrases. Le

---

4 Plus d'informations sur le projet Alector : https://alectorsite.wordpress.com/

5 https://korp.csc.fi/download/opusparcus/

6 OpenSubtitles2016 est un sous-ensemble de la collection OPUS ("... *the open parallel corpus*") http://opus.nlpl.eu/, et fournit un grand nombre de corpus parallèles alignés sur les phrases dans 65 langues.

processus de simplification peut ainsi être considéré comme un processus de reconnaissance de paraphrase dirigée ou d'implication textuelle, avec une contrainte de lisibilité sur le texte simplifié. La relation de simplification est donc asymétrique, contrairement à la paraphrase, elle se rapproche ainsi de l'implication textuelle. Les opérations de simplification visent à préserver la plupart des informations contenues dans le texte. Par conséquent, les informations périphériques sont supprimées, d'où le fait que les techniques de résumé jouent un rôle important dans la SAT.

Enfin, au cours des dernières années, la disponibilité des corpus parallèles de textes originaux et simplifiés a rendu possible un ensemble d'approches permettant d'apprendre différents types d'opérations de simplification à partir de corpus. Cependant, la SAT peut être appréhendée avec des méthodes de traduction automatique et d'apprentissage automatique dont les modèles statistiques sont construits à partir de corpus parallèles de textes originaux et simplifiés (Zhu et al., 2010; Specia, 2010; Woodsend et Lapata, 2011). Notamment, la disponibilité du corpus PWKP (*Parallel Wikipedia Simplification Corpus*) constitué par Zhu et ses collaborateurs (2010) a eu un impact considérable. Sa taille et sa disponibilité en ont fait le jeu de données de référence des travaux de simplification pour l'anglais. Il se compose d'un texte aligné de Wikipédia anglais[7] et de son équivalent dans Simple English Wikipedia[8]. L'ensemble de données contient 108 016 paires de phrases, avec 25,01 mots en moyenne par phrase 'complexe' et 20,87 mots par phrase simple.
Cependant, Xu et ses collaborateurs (2015) ont présenté une prise de position, dans laquelle ils décrivent plusieurs lacunes de cette ressource et contient de bruits. Ils ont approfondi cette étude par une annotation manuelle de 200 alignements de phrases choisies aléatoirement. Ils ont montré que seuls 50% des paires de phrases correspondent à une simplification. Afin de répondre au problème du bruit présent dans les alignements du corpus Wikipédia anglais, les auteurs ont introduit une nouvelle ressource : le corpus Newsela[9] (Xu et al., 2015). Dans ce travail, nous étudions les différentes transformations caractérisant le passage d'un niveau de simplification à l'autre sur un sous-ensemble de textes, en nous intéressant plus particulièrement aux transformations syntaxiques.

2.2 Le corpus Newsela

Newsela est un jeu de données pour la simplification de texte qui a comme avantage le fait de proposer différentes versions de simplification à partir d'un texte original (la plupart des corpus parallèles existants, mentionnés plus haut, sont des corpus parallèles des textes originaux et simplifiés). Il s'agit d'un corpus d'articles de presse réécrits par des éditeurs professionnels. Le public cible considéré était les enfants de différents niveaux scolaires. Le corpus contient 1.130 articles de presse, chacun d'eux réécrit 4 fois (tableau 1) pour les enfants de différents niveaux, obtenant quatre versions de simplification (simp-1 est la moins simple et simp-4 la plus simple).

Texte original	**The athletic shoe and apparel maker** said Thursday it will provide free design resources to schools looking to shelve Native American mascots, nicknames, imagery or symbolism. **The German company** also pledged to provide financial support to ensure the cost of changing is not prohibitive.
Simp_1	**The athletic shoe and apparel maker** said Thursday it will provide free design resources to schools looking to give up their Native American mascots, nicknames, imagery or symbolism. **It** also pledged to provide financial help to ensure the cost of changing is not excessive.
Simp_2	**Adidas** officials said Thursday it will help the schools create new mascots, nicknames, images or symbols. **It** also promised to help cover the cost to make sure

7 https://www.wikipedia.org/
8 https://simple.wikipedia.org/
9 https://newsela.com/data/

	that the change is not too expensive.
Simp_3	On Thursday, **Adidas** offered to help high schools change their mascots. A mascot is usually a person or animal. It represents a group or school, and many people think it brings good luck. Most sports teams have a mascot. **The shoe and clothing company** will also help to pay for the change. New uniforms, mascots and signs can be expensive for schools.
Simp_4	**Adidas** said it will help schools make new mascots. A mascot can be a person or an animal. Most sports teams have a mascot. **Adidas** will help schools design new uniforms. It will also help them to design new logos. Logos are the pictures on uniforms or signs. It costs a great deal of money to change logos and mascots. Adidas will help schools pay for it.

TABLE 1. Exemple de phrases écrites à plusieurs niveaux de complexité de texte à partir de l'ensemble de données Newsela.

3 Méthodologie

Comme nous venons de le voir, le corpus Newsela est un corpus parallèle dédié à la simplification de textes, proposant pour un texte donné, quatre niveaux de simplification. Nous nous intéressons dans cet article aux différentes transformations caractérisant le passage d'un niveau de simplification à l'autre. Xu et ses collaborateurs (2015) ont effectué une analyse systématique de l'ensemble du corpus en se focalisant sur l'aspect lexical. Notre but est d'analyser ce corpus en nous focalisant sur les aspects syntaxiques, c'est-à-dire, étudier les changements syntaxiques qui ont été faits lors du passage d'un niveau de difficulté à un autre plus simple. Une analyse qualitative a été réalisée afin de cibler différents types d'opérations de simplification.

Le travail que nous décrivons dans cette proposition est basé sur un sous-ensemble de textes composés de 107 phrases tirées de 6 textes choisis d'une façon aléatoire du corpus Newsela, en nous intéressant plus particulièrement aux transformations syntaxiques. Nous avons aligné ces originaux avec les 4 niveaux de difficulté et nous avons repéré les opérations effectuées au niveau syntaxique.

3.1 Analyse quantitative du corpus Newsela

Le Tableau 1 récapitule le nombre de tokens et de phrases dans l'ensemble des 6 textes étudiés. Ce tableau montre le nombre total de phrases et de mots et la longueur moyenne de la phrase (en mots) des textes simplifiés originaux et les 4 niveaux de simplification : de l'original au Simp_1, du Simp_1 au Simp_2, du Simp_2 au Simp_3 et du Simp_3 au Simp_4.

Il y a une réduction importante au niveau de la longueur du texte en passant du niveau 2 au niveau 3 et une autre plus grande (27 %) en passant du niveau 3 au niveau 4, ce qui était attendu pour ce type de public (les apprenants d'une langue seconde). Dans ce type de corpus, des ajouts sont considérés comme utiles pour améliorer la lecture et la compréhension de textes mais privilégiant la suppression des informations supplémentaires et redondantes en gardant toujours des phrases courtes.

	Tokens		Phrases		Tokens/phrase	
	Total	Réduction	Total	Augmentation	Moyenne	Réduction
Textes originaux	4.866		107		45,5	
Simp_1	4.577	6 %	118	10 %	38,8	15 %
Simp_2	4.358	5 %	165	40 %	26,4	32 %
Simp_3	3.768	14 %	201	22 %	18,7	29 %
Simp_4	2.740	27 %	192	-4 %	14,3	24 %

TABLE 1 : Nombre de tokens et de phrases dans 6 textes du corpus Newsela : original → Simp_1, Simp_1 → Simp_2, Simp_2 → Simp_3 et Simp_3 → Simp_4

Le pourcentage d'augmentation du nombre de phrases est clairement important en passant du niveau 1 au niveau 2 (40 %). En comparant le nombre de phrases entre le texte original et sa version la plus simplifiée (Simp_4), le nombre augmente de 79 %. Ces différences s'expliquent essentiellement par les opérations de découpage des phrases longues appliquées. Le nombre de mots par phrase diminue en passant d'un niveau à un autre, ce qui revient à des phrases plus courtes qui maintiennent la structure SVO et qui présentent une seule idée par phrase.

Nous avons utilisé l'outil CollateX[10] (Dekker et Middell, 2011) pour repérer les transformations (découpage de phrases, suppression et substitution morpho-syntaxiques, insertion d'informations, réorganisation et fusion) effectuées dans les versions simplifiées par rapport au texte original (Original Simp_1, du Simp_1 au Simp_2, du Simp_2 au Simp_3 et du Simp_3 au Simp_4). CollateX est un outil utilisé dans les humanités numériques qui implémente des algorithmes d'alignement et fournit une visualisation statique pour les graphiques de variantes de texte. Dans cet outil, quatre étapes de base sont définies et appliquées dans l'ordre et/ou de manière itérative. La première est la tokenisation des textes numériques à comparer. La deuxième étape est l'alignement des tokens de différents textes et implication des opérations d'édition. La troisième étape est l'analyse de l'alignement calculé, les opérations d'édition étant désormais qualifiées (par exemple, suppression, ajout ou déplacement). La quatrième et dernière étape est la sortie ou visualisation des résultats. La figure 1 montre la fréquence des différentes opérations de transformation dans les 6 textes du corpus.

De ces résultats d'analyse, nous pouvons déduire les spécificités de chaque niveau de simplification :

- Le ***niveau 1*** privilégie la réorganisation de la phrase et substitution morpho-syntaxique : des clauses peuvent être échangées afin que la présentation de l'information soit plus lisible. De plus, lorsque des structures complexes ne sont pas supprimées, elles sont généralement déplacées pour faciliter la compréhension. Les opérations de remplacements et de réorganisations représentent respectivement 35 % et 32 % des transformations syntaxiques dans ce niveau de difficulté.

[10] https://collatex.net/demo/

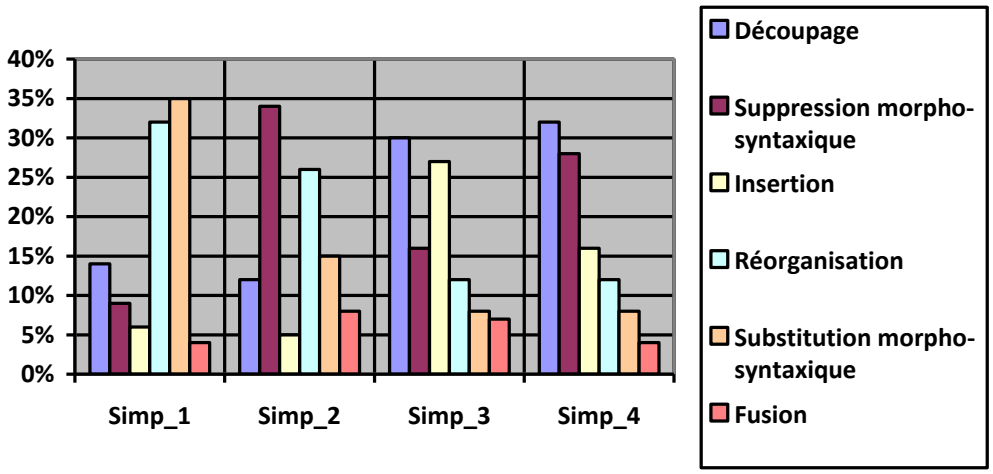


FIGURE 1 : Fréquence des opérations de transformation syntaxique dans les 6 textes du corpus Newsela

- Le ***niveau 2*** privilégie les suppressions morpho-syntaxiques (34 %) : les informations d'importance secondaire sont supprimées des phrases.

- Le ***niveau 3*** privilégie le découpage de phrases et les insertions : les auteurs choisissent de diviser des phrases trop longues (quand il y a coordination ou signe de ponctuation). Le découpage et les insertions représentent respectivement 30 % et 27% des transformations syntaxiques dans ce niveau de complexité.

- Le ***niveau 4*** privilégie les découpage et suppressions morpho-syntaxiques : les découpages des phrases complexes représentent 32 % des transformations et les suppressions 28%.

4 Analyse des opérations de simplification

Nous avons observé manuellement 107 phrases alignées du corpus dans ses 4 niveaux de difficultés afin d'en dégager les différents phénomènes intervenant dans la simplification de texte.

Les opérations de simplification présentées dans le corpus Newsela s'appliquent à plusieurs niveaux : lexical, morpho-syntaxique et discursif, ce qui reste classique dans le cadre de la simplification de textes (Gala et al., 2018). Dans cet article, nous présentons uniquement une caractérisation du niveau syntaxique. Sur la base de différentes classifications des opérations de simplification proposée dans la littérature (Brunato et al., 2015 ; Bott et Saggion, 2014 ; Coster et Kauchak, 2011 ; Caseli et al., 2009, Medero et Ostendorf, 2011 et Zhu et al., 2010, Gala et al. 2020b), nous avons identifié six classes principales d'opérations observées dans le corpus Newsela, à savoir : (4.1) des découpages, (4.2) des fusions, (4.3) des réorganisations, (4.4) des insertions (rajouts), (4.5) des suppressions et (4.6) des substitutions.

4.1 Découpage de phrases

C'est l'opération la plus fréquente en SAT, pour les applications à la fois humaines et automatiques. En règle générale, le découpage supprime les conjonctions, les deux points, les points-virgules, les énumérations et les appositives afin d'obtenir deux phrases indépendantes. Dans (1) et (2), nous donnons deux exemples de découpage. Nous pouvons observer qu'elles s'appliquent à partir du premier niveau de simplification.

(1) *Originale*. The advocacy group says about a dozen schools have dropped Native American mascots over the past two years **and** an additional 20 are considering a change.
Simpl_1. The advocacy group says about a dozen schools have dropped Native American mascots over the past two years. An additional 20 are considering a change.
Simpl_2. The advocacy group said about a dozen schools have dropped these mascots over the past two years. An additional 20 are considering a change.
Simpl_3. About a dozen of the schools decided to pick new mascots in the last two years. Another 20 are thinking about it.
Simpl_4. In the last two years, about 12 schools stopped using them. Another 20 are thinking about it.

(2) *Originale.* **The company, which** has its North American headquarters in Portland, Oregon, also said it will be a founding member of a coalition that addresses Native American mascots in sports.
Simpl_1. Adidas also said it will be a founding member of a coalition that addresses the problem of Native American mascots in sports**. The German company** has its North American headquarters in Portland, Oregon.
Simpl_2. Adidas also said it will be a founding member of a group that deals with the problem of Native American mascots in sports. The German company makes shoes and clothing. **Its** North American headquarters is in Portland, Oregon.
Simpl_3. Adidas said it will help start a new group. It will deal with the problem of Native American mascots in sports.
Simpl_4. Phrases Supprimées.

Les appositions ne sont pas supprimées, elles sont transformées en phrases indépendantes :

(3) *Originale***.** Eric Liedtke**, Adidas head of global brands,** traveled to the conference. He said sports must be inclusive.
Simpl_1. Eric Liedtke, **Adidas head of global brands**, said sports must be inclusive.
Simpl_2. Eric Liedtke **is** the Adidas head of global brands. He said that sports must include everyone.
Simpl_3. Eric Liedtke, **who works for Adidas**, attended the Tribal Nations conference. Sports must include everyone, he said.
Simpl_4. Eric Liedtke **works for Adidas**. Sports must include everyone, he said.

Les exemples 2 et 3 montrent que les opérations ne sont pas indépendantes les uns des autres, puisque la substitution d'une phrase entraîne un changement syntaxique.

4.2 Fusion de phrases

Cette opération est conçue comme l'inverse de la division, c'est l'opération par laquelle deux (ou plusieurs) phrases originales sont fusionnées en une phrase simplifiée unique. Une telle

transformation est moins fréquente que la division des phrases (< 8% dans tous les niveaux de simplification).

(4) *Originale*. Eric Liedtke, Adidas head of global brands, traveled to the conference. **He said sports must be inclusive.**
Simpl_1. Eric Liedtke, Adidas head of global brands, **said sports must be inclusive.**

4.3 Réorganisation

Lorsque certaines structures complexes ne sont pas supprimées, elles sont souvent déplacées ou modifiées dans le texte dans le but de maintenir une structure SVO (Gala et al., 2020b). Cette opération marque le changement de position de mots entre la phrase d'origine et son équivalent simplifié (ex. 5).

(5) *Originale*. **According to** the group Change the Mascot, there are about 2,000 schools nationwide that have Native American mascots.
Simpl_1. **According to** the group Change the Mascot, there are about 2,000 schools nationwide that have Native American mascots.
Simpl_2. About 2,000 schools nationwide have Native American mascots, **according to** the group Change the Mascot.
Simpl_3. Change the Mascot is a group that wants schools to stop using Native American mascots. About 2,000 American schools have them, **the group said**.
Simpl_4. Change the Mascot is a Native American group. **It wants** schools to drop Native American mascots. About 2,000 American schools have these mascots.

4.4 Insertion

Le processus de simplification peut entraîner une phrase plus longue, en raison de rajout de mots ou d'expressions qui fournissent des informations de clarification (ex. 6). Nous avons observé un seul type d'informations pour marquer les insertions : le rajout d'explications et de définitions. Ce procédé est approprié pour les corpus destinés à des apprenants normo-lecteurs ou à des adultes illettrés ; il ne l'est pas pour d'autres types de public cible, par exemple les enfants faibles-lecteurs ou dyslexiques (Rello, 2014).

(6) *Originale*. The NFL's Washington **Redskins** have resisted appeals by Native American and civil rights groups to change their name and mascot.
Simpl_1. The NFL's Washington **Redskins** have resisted appeals by Native American and civil rights groups to change the name and mascot.
Simpl_2. Native American and civil rights groups have asked NFL's Washington **Redskins** to change their name and mascot. The football team has refused.
Simpl_3. They have repeatedly asked the Washington Redskins football team to change its name and mascot. **Redskins is an old word for Native Americans, who feel that it is unkind**. The football team has refused to change its name.
Simpl_4. Americans have asked the Washington Redskins to change the team's name. **The Redskins is a football team. Redskins is a very old word for Native Americans. It is not a nice word.** The football team has refused again and again.

4.5 Suppression morpho-syntaxique

Les informations secondaires ou redondantes, généralement considérées comme supplémentaires au niveau syntaxique, ne sont pas incluses dans les textes simplifiés (ex. 7 à 10). Un texte devrait être simplifié en éliminant les informations redondantes. Les phrases simplifiées contiennent moins d'adverbes ou d'adjectifs que les phrases originales. Certains adverbes et adjectifs, entre autres, sont omis. Nous proposons six types d'informations qui peuvent être supprimées : les informations entre parenthèses, les exemples, les constructions appositives, certains modifieurs, quelques relatives ainsi que les expressions temporelles et locatives dans certains cas.

(7) *Originale*. Some colleges kept their nicknames by obtaining permission from tribes, **including the Florida State Seminoles and the University of Utah Utes.**
Simpl_1. Several colleges kept their nicknames by obtaining permission from tribes, **such as the Florida State Seminoles and the University of Utah Utes.**
Simpl_2. Some colleges kept their nicknames by getting permission from tribes. Two teams that received permission were the **Florida State Seminoles and the University of Utah Utes**.
Simpl_3. Some colleges were able to keep their names, though. They received permission from tribes.
Simpl_4. Some colleges kept their names, though. They asked for permission from Native American groups.

(8) *Originale*. […] Ray Halbritter applauded Adidas' move in a **joint statement**.
Simpl_1. […] Ray Halbritter applauded Adidas' announcement in a **joint statement**.
Simpl_2. […] Ray Halbritter applauded Adidas' announcement in a **statement**.

(9) *Originale*. **In 2005**, the NCAA warned schools that they would face sanctions if they didn't change Native American logos or nicknames.
Simpl_1. **In 2005**, the NCAA warned schools that they would face sanctions if they did not change Native American logos or nicknames.
Simpl_2. **In 2005**, the National Collegiate Athletic Association (NCAA) warned colleges that they would face penalties if they did not change Native American logos or nicknames.
Simpl_3. **In 2005**, the National Collegiate Athletic Association (NCAA) told colleges to stop using Native American mascots.
Simpl_4. It told the colleges to get new mascots. If not, the colleges could be punished.

(10) *Originale*. Adidas announced the initiative in conjunction with the White House Tribal Nations Conference on Thursday **in Washington**.
Simpl_1. Adidas announced the initiative in conjunction with the White House Tribal
Simpl_2. Nations Conference on Thursday **in Washington**, **D.C**.
Adidas announced the project as the White House Tribal Nations Conference met **in Washington, D.C.**, on Thursday.
Simpl_3. The White House Tribal Nations Conference took place on Thursday.

4.6 Substitution morpho-syntaxique

Nous avons observé des substitutions de nature morpho-syntaxique : transformer les phrases passives en phrases actives, privilégier les propositions positives à la place des propositions négatives, ainsi que les formes personnelles à la place des formes impersonnelles.

(11) *Originale*. The NFL's Washington Redskins have resisted appeals **by** Native American and civil rights groups to change their name and mascot.
Simpl_4. Native Americans have asked the Washington Redskins to change the team's name.

(12) *Originale*. "Today's announcement is a great way for us to offer up our resources to schools that want to do what's right — to administrators, teachers, students and athletes who want to make a difference in their lives and in their world," Liedtke said in a statement to The Associated Press. "Our intention is to help break down any barriers to change — change that can lead to a more respectful and inclusive environment for all American athletes."
Simpl_3. Liedtke said that many schools want to do what is right. Teachers and students want to make a difference in their lives and in the world. Adidas wants to make it easier for them, he said. He said the new school mascots will respect all American athletes.

En plus de ces variations linguistiques, les versions simplifiées de Newsela présentent des variations typographiques[11] notamment avec des variations des nombres. Ce procédé est assez courant pour alléger la charge cognitive pendant la lecture (spécialement pour les lecteurs en difficulté de lecture, voir Rello, 2014 ; Gala et al. 2020a) :

(13) *Originale***.** The advocacy group says about a **dozen** schools have dropped Native American mascots over the past two years and an additional 20 are considering a change.
Simpl_4. In the last two years, about **12** schools stopped using them. Another 20 are thinking about it.

5 Conclusion

La disponibilité de corpus parallèles monolingues est fondamentale pour à la recherche en simplification automatique du texte (SAT). Ces corpus constituent, malgré leur rareté et la difficulté de leur construction, les corpus les plus appropriés et adaptés pour l'étude de la simplification de SAT. La majorité des méthodes de simplification syntaxique reposent sur un ensemble de règles de transformation définies manuellement pour être appliquées aux phrases. Dans Newsela, l'existence de quatre versions correspondant à quatre niveaux de difficulté rend le corpus intéressant à étudier dans le but de comprendre les transformations et pouvoir les implémenter plus tard dans un système de SAT.

Dans cet article, nous avons présenté une étude de transformations syntaxiques qui nous permettent de décrire 6 textes originaux en anglais et leurs simplifications. Notre étude du corpus Newsela est à la fois qualitative et quantitative, sur les transformations appliquées lors du passage d'un niveau de difficulté à un autre. Nous nous sommes basées sur un sous-ensemble de textes restreint de Newsela. Dans le but de généraliser nos résultats, il sera intéressant de mener des expériences sur la totalité du corpus.

Par la suite, notre objectif sera d'appréhender la simplification automatique de textes selon une approche à base de représentations sémantiques, en utilisant le formalisme sémantique UCCA (*Universal Cognitive Conceptual Annotation* ; Abend et Rappoport, 2013 ; Sulem et al., 2018). Les

[11] Bouamor (2012) a défini les variations typographiques dans la catégorisation des paraphrases sous-phrastiques qui peut être aussi une classe définie pour la catégorisation de la simplification syntaxique.

informations sémantiques sont fondamentales d'où notre intérêt à déterminer automatiquement, au moyen d'un formalisme, quelles informations sont secondaires. Ce faisant elles pourront être supprimées et il sera alors possible de rendre les informations primordiales plus visibles aux lecteurs en difficulté (travaux en cours).

Remerciement

Je tiens à remercier Núria GALA (LPL) et Bernard ESPINASSE (LIS) pour leur aide précieuse et pour leurs contributions à la réalisation de ce travail.

Références bibliographiques

ABEND, O., & RAPPOPORT, A. (2013, August). Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 228-238).

ALUÍSIO, S. M., SPECIA, L., PARDO, T. A., MAZIERO, E. G., CASELI, H. M., & FORTES, R. P. (2008, September). A corpus analysis of simple account texts and the proposal of simplification strategies: first steps towards text simplification systems. In *Proceedings of the 26th annual ACM international conference on Design of communication* (pp. 15-22).

ALVA-MANCHEGO, F., BINGEL, J., PAETZOLD, G., SCARTON, C., & SPECIA, L. (2017, November). Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 295-305).

BOTT, S., & SAGGION, H. (2014). Text simplification resources for Spanish. *Language Resources and Evaluation*, *48*(1), 93-120.

BOUAMOR, H. (2012). *Étude de la paraphrase sous-phrastique en traitement automatique des langues* (Doctoral dissertation).

BROUWERS, L., BERNHARD, D., LIGOZAT, A. L., & FRANÇOIS, T. (2012, June). Simplification syntaxique de phrases pour le français (Syntactic Simplification for French Sentences) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN* (pp. 211-224).

BROUWERS, L., BERNHARD, D., LIGOZAT, A. L., & FRANÇOIS, T. (2014, April). Syntactic sentence simplification for French. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)* (pp. 47-56).

BRUNATO, D., DELL'ORLETTA, F., VENTURI, G., & MONTEMAGNI, S. (2015, June). Design and annotation of the first Italian corpus for text simplification. In *Proceedings of The 9th Linguistic Annotation Workshop* (pp. 31-41).

BRUNATO, D., CIMINO, A., DELL'ORLETTA, F., & VENTURI, G. (2016, November). Paccss-it: A parallel corpus of complex-simple sentences for automatic text simplification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 351-361).

BURSTEIN, J., SHORE, J., SABATINI, J., LEE, Y. W., & VENTURA, M. (2007, April). The automated text adaptation tool. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)* (pp. 3-4).

BURSTEIN, J. (2009, March). Opportunities for natural language processing research in education. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 6-27). Springer, Berlin, Heidelberg.

CANNING, Y., TAIT, J., ARCHIBALD, J., & CRAWLEY, R. (2000, September). Cohesive generation of syntactically simplified newspaper text. In *International Workshop on Text, Speech and Dialogue* (pp. 145-150). Springer, Berlin, Heidelberg.
CARROLL, J., MINNEN, G., CANNING, Y., DEVLIN, S., & TAIT, J. (1998, July). Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology* (pp. 7-10).
CASELI, H. M., PEREIRA, T. F., SPECIA, L., PARDO, T. A., GASPERIN, C., & ALUÍSIO, S. M. (2009). Building a Brazilian Portuguese parallel corpus of original and simplified texts. *Advances in Computational Linguistics, Research in Computer Science*, *41*, 59-70.
CHANDRASEKAR, R., DORAN, C., & SRINIVAS, B. (1996, August). Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2* (pp. 1041-1044). Association for Computational Linguistics.
COSTER, W., & KAUCHAK, D. (2011, June). Learning to simplify sentences using wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation* (pp. 1-9). Association for Computational Linguistics.
CREUTZ, M. (2018). Open Subtitles Paraphrase Corpus for Six Languages. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). *arXiv preprint arXiv:1809.06142.*
CROSSLEY, S. A., LOUWERSE, M. M., MCCARTHY, P. M., & MCNAMARA, D. S. (2007). A linguistic analysis of simplified and authentic texts. *The Modern Language Journal*, *91*(1), 15-30.
DEKKER, R. H. ET MIDDELL, G. (2011). Computer-supported collation with CollateX : Managing textual variance in an environment with varying requirements. *Supporting Digital Humanities*, (pp.17–18).
DEVLIN, SIOBHAN AND JOHN TAIT. (1998). The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases* (pp. 161–173).
EVANS, R., ORASAN, C., & DORNESCU, I. (2014). An evaluation of syntactic simplification rules for people with autism. Association for Computational Linguistics.
FENG, L. (2008). Text simplification: A survey. *The City University of New York, Tech. Rep*.
GALA, N., FRANÇOIS, T., JAVOUREY-DREVET, L., & ZIEGLER, J. C. (2018). La simplification de textes, une aide à l'apprentissage de la lecture. *Langue française*, (3), 123-131.
GALA, N., TACK, A., JAVOUREY-DREVET, L., FRANÇOIS, T., & ZIEGLER, J. C. (2020a). Alector: A Parallel Corpus of Simplified French Texts with Alignments of Misreadings by Poor and Dyslexic Readers. In *Language Resources and Evaluation for Language Technologies (LREC)*.
GALA N., TODIRASCU, A., BERNHARD, D., WILKENS, R. ET MEYER, J.-P. (2020b) Transformations syntaxiques pour une aide à l'apprentissage de la lecture : typologie, adéquation et corpus adaptés. Actes du *Congrès Mondial de Linguistique Française* (CMLF 2020). Montpellier, France.
GASPERIN, C., SPECIA, L., PEREIRA, T., & ALUÍSIO, S. (2009). Learning when to simplify sentences for natural text simplification. In *Proceedings of ENIA*, 809-818.
HASLER, E., DE GISPERT, A., STAHLBERG, F., WAITE, A., & BYRNE, B. (2017). Source sentence simplification for statistical machine translation.
INUI, K., FUJITA, A., TAKAHASHI, T., IIDA, R., & IWAKURA, T. (2003, July). Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16* (pp. 9-16). Association for Computational Linguistics.
KOPTIENT, A., CARDON, R. ET GRABAR, N. (2019) Simplification-induced transformations: typology and some characteristics. In *Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics*, Florence, Italy (pp. 309-318).
LISON, P. AND TIEDEMANN, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016).*

MAX, A. (2005). Simplification interactive pour la production de textes adaptés aux personnes souffrant de troubles de la compréhension. In *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*.
MAX, A. (2006). Writing for language-impaired readers. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 567-570). Springer, Berlin, Heidelberg.
MEDERO, J., & OSTENDORF, M. (2011). Identifying targets for syntactic simplification. In *Speech and Language Technology in Education*.
NEWSELA. (2016). Newsela article corpus. https://newsela.com/data. Version : 2016-01-29.
PETERSEN, S. E., & OSTENDORF, M. (2007). Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*.
RELLO, L., BAYARRI, C., GÒRRIZ, A., BAEZA-YATES, R., GUPTA, S., KANVINDE, G., ... & TOPAC, V. (2013, May). DysWebxia 2.0! More accessible text for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility* (pp. 1-2).
RELLO, L. (2014). *DysWebxia: a text accessibility model for people with dyslexia* (Doctoral dissertation, Universitat Pompeu Fabra).
SAGGION, H. (2017). Automatic Text Simplification: Synthesis Lectures on Human Language Technologies, vol. 10 (1). California, Morgan & Claypool Publishers.
SIDDHARTHAN, A. (2006). Syntactic simplification and text cohesion. *Research on Language and Computation*, *4*(1), 77-109.
SILVEIRA, S. B., & BRANCO, A. (2012). Enhancing multi-document summaries with sentence simplification. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
SPECIA, L. (2010, April). Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language* (pp. 30-39). Springer, Berlin, Heidelberg.
SPECIA, L., JAUHAR, S. K., & MIHALCEA, R. (2012, June). Semeval-2012 task 1: English lexical simplification. In *Proceedings of 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* (pp. 347-355). Association for Computational Linguistics.
SULEM, E., ABEND, O., & RAPPOPORT, A. (2018). Simple and effective text simplification using semantic and neural methods. *arXiv preprint arXiv:1810.05104.*
VAJJALA, S., & MEURERS, D. (2014). Readability assessment for text simplification: From analysing documents to identifying sentential simplifications. *ITL-International Journal of Applied Linguistics*, *165*(2), 194-222.
VANDERWENDE, L., SUZUKI, H., BROCKETT, C., & NENKOVA, A. (2007). Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, *43*(6), 1606-1618.
WOODSEND, K., & Lapata, M. (2011, July). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 409-420). Association for Computational Linguistics.
XU, W., CALLISON-BURCH, C., & NAPOLES, C. (2015). Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, *3*, 283-297.
ZHU, Z., BERNHARD, D., & GUREVYCH, I. (2010, August). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1353-1361). Association for Computational Linguistics.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

La désambiguïsation des abréviations du domaine médical

Anaïs Koptient[1]
(1) CNRS, UMR 8163, Univ. Lille, UMR 8163 - STL - Savoirs Textes Langage, F-59000 Lille, France
anais.koptient.etu@univ-lille.fr

RÉSUMÉ

Les abréviations, tout en étant répandues dans la langue, ont une sémantique assez opaque car seulement les premières lettres sont transparentes. Cela peut donc empêcher la compréhension des abréviations, et des textes qui les contiennent, par les locuteurs. De plus, certaines abréviations sont ambiguës en ayant plusieurs sens possibles, ce qui augmente la difficulté de leur compréhension. Nous proposons de travailler avec les abréviations de la langue médicale dans un cadre lié à la simplification automatique de textes. Dans le processus de simplification, il faut en effet choisir la forme étendue des abréviations qui soit correcte pour un contexte donné. Nous proposons de traiter la désambiguïsation d'abréviations comme un problème de catégorisation supervisée. Les descripteurs sont construits à partir des contextes lexical et syntaxique des abréviations. L'entraînement est effectué sur les phrases qui contiennent les formes étendues des abréviations. Le test est effectué sur un corpus construit manuellement, où les bons sens des abréviations ont été définis selon les contextes. Notre approche montre une F-mesure moyenne de 0,888 sur le corpus d'entraînement en validation croisée et 0,773 sur le corpus de test.

ABSTRACT

Disambiguation of abbreviations from the medical domain.

Abbreviations, although commonly used, have quite opaque semantics because only their first letters are transparent. This may prevent from understanding of abbreviations, and of texts they occur within, by speakers. Besides, some abbreviations are ambiguous and have more than one meaning, which increases their understanding difficulty. We propose to work with abbreviations from the medical domain as part of the automatic text simplification. During the simplification process, it is indeed necessary to chose the right expanded form of abbreviations satisfying a given context. We propose to address disambiguation of abbreviations as supervised categorization problem. Descriptors are built from lexical and syntactic contexts of the abbreviations. Training is done on sentences containing expanded forms of the abbreviations. Test is done on corpus built manually, in which the correct senses of abbreviations have been defined according to their contexts. The average F-measure of our approach is 0.888 in cross-validation on the training corpus and 0.773 on the test corpus.

MOTS-CLÉS : Désambiguïsation sémantique, domaine biomédical, abréviations, simplification.

KEYWORDS: Word sense disambiguation, Medical domain, Abbreviations, Simplification.

1 Introduction

Les abréviations sont assez répandues dans les informations et les situations qui nous entourent. Les quelques exemples en (1) illustrent la variété de ces situations.

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 151–163.
HAL : hal-02786196.

(1) *CAF => Caisse des Allocations Familiales*
PC => Personal Computer
ADP => Aéroports de Paris
TGV => Train à grande vitesse
AVC => Accident Vasculaire Cérébral
IRM => Imagerie par Résonance Magnétique
DP => Dialyse Péritonéale
ACP => analgésie contrôlée par le patient
AC => anhydrase carbonique

Notre compétence linguistique face aux abréviations varie en fonction de la nature de ces abréviations. Ainsi, la signification de certaines abréviations peut nous être connue du fait de leur fréquence d'emploi dans la langue et de leur implication dans le quotidien, comme c'est le cas de *CAF, PC, TGV, ADP* ou même *AVC* et *IRM*. Avec d'autres abréviations, la compréhension est beaucoup moins aisée, comme par exemple avec *DP, ACP* ou *AC*. Dans ces cas, il est nécessaire de disposer de la forme étendue des abréviations pour mieux les comprendre. En effet, les abréviations ont une sémantique très opaque car seulement les premières lettres sont transparentes alors que le reste des mots ne l'est pas. De plus, les abréviations et les mots qui les composent peuvent être spécifiques aux domaines de spécialité, comme le domaine médical dans les exemples que nous utilisons. La sémantique devient alors encore plus opaque. Pour aider le lecteur à bien comprendre la signification des abréviations, et des textes qui les comportent, il est nécessaire de fournir au moins les formes étendues des abréviations. Notons que, dans ce qui suit, nous utilisons de manière équivalente les termes suivants : forme étendue, forme développée, développement et sens des abréviations.

La simplification, automatique ou manuelle, de textes a justement pour objectif de rendre un texte plus lisible et compréhensible. Ainsi, différents guides d'aide à la rédaction de documents simples et accessibles (OCDE, 2015; Ruel J. & L., 2011; UNAPEI, 2019) préconisent, entre autres, de fournir les formes étendues des abréviations. Cela correspond au cadre dans lequel s'inscrit notre travail : simplifier automatiquement les documents en langue française pour les rendre plus faciles à comprendre. Nous nous intéressons ici plus particulièrement à l'explicitation et la simplification d'abréviations. Une autre particularité est que la simplification est effectuée avec les documents techniques du domaine médical. Ce domaine, tout en touchant intimement à notre vie, manipule typiquement de nombreux termes et abréviations spécialisés.

La simplification d'abréviations repose sur la disponibilité et l'exploitation de ressources dédiées, où les abréviations sont associées avec leurs formes étendues, comme dans les exemples en (1). Cependant, de nombreuses abréviations peuvent être ambiguës et avoir plusieurs développements possibles. Par exemple, dans la langue générale, l'abréviation *PC* peut signifier *Personal Computer* mais aussi *Parti Communiste*, et seul le contexte pourra indiquer quel sens et quelle forme étendue correspondante sont corrects pour une occurrence donnée de cette abréviation. La même situation se vérifie dans les langues de spécialité : les abréviations techniques peuvent également être ambiguës et avoir plusieurs développements possibles. C'est notamment le cas des abréviations en (2).

(2) *DP* : *Dialyse Péritonéale*, *Dilatation Pneumatique*, *Dysménorrhée Primaire* ;
ACP : *amplification en chaîne par polymérase*, *analgésie contrôlée par le patient* ;
AC : *ablation par cathéter*, *âge corrigé*, *acétate de cyprotérone*, *anhydrase carbonique*, *anthracycline et cyclophosphamide* ;
ADP : *accès douloureux paroxystiques*, *adénosine diphosphate*.

Le nombre d'abréviations ambiguës est potentiellement élevé et peut donc concerner un nombre assez important de phrases. Le fait qu'une abréviation ait plusieurs développements possibles devient donc problématique, car le système de simplification doit sélectionner la forme développée correcte pour une abréviation étant donné le contexte. Il est donc nécessaire d'effectuer la désambiguïsation pour simplifier une phrase donnée correctement.

Dans ce qui suit, nous présentons d'abord un état de l'art sur la désambiguïsation de mots et termes (section 2). Ensuite, nous décrivons notre démarche pour désambiguïser les abréviations du domaine médical en français (section 3). Les résultats sont présentés et discutés dans la section 4. Nous concluons avec quelques pistes d'amélioration et les perspectives pour les travaux futurs.

2 État de l'art

L'intérêt pour la désambiguïsation sémantique a été manifesté très tôt par la communauté de TAL : dès l'arrivée des premiers programmes informatiques au début des années 50 (Ide & Véronis, 1998). Actuellement, la désambiguïsation est utilisée en pré-traitement de plusieurs autres programmes et applications, comme par exemple :

— la traduction automatique car un mot de la langue source peut avoir plusieurs traductions possibles dans la langue cible, en fonction de sa sémantique (Vickrey *et al.*, 2005; Miháltz, 2005; Li & Li, 2004; Specia, 2005; Lim & Tang, 2004; Marvin & Koehn, 2018; Tang *et al.*, 2018; Parameswarappa & Narayana, 2011; Brown *et al.*, 1991) ;

— l'extraction d'information car, lors de la recherche de mots-clés, il est important de pouvoir éliminer les mots-clés qui n'ont pas le sens recherché (Stokoe *et al.*, 2003; Zhong & Ng, 2012; Whaley, 1999; Krovetz, 2002; Stokoe & Tait, 2002) ;

— l'analyse grammaticale lors de l'annotation en parties du discours, par exemple pour éviter d'annoter de manière erronée un mot dont l'homonyme n'a pas la même partie du discours (Bikel, 2000).

Les travaux que nous mentionnons ici concernent la désambiguïsation effectuée essentiellement sur les données de la langue médicale, y compris sur les abréviations du domaine médical, car c'est dans ce domaine que nous positionnons notre travail.

Pour la présentation de travaux existants, nous différencions les méthodes non supervisées (section 2.1) et les méthodes supervisées (section 2.2). L'avantage de méthodes non supervisées est qu'elles ne demandent pas de gros corpus annotés pour l'entraînement de systèmes et sont donc plus faciles à mettre en place. En revanche, elles obtiennent théoriquement de moins bons résultats que les méthodes supervisées car elles utilisent moins de connaissances et disposent d'une plus petite fraction de la vérité de terrain (Zhou & Han, 2005). Nous verrons cependant que les chercheurs mettent en place différentes approches afin de réduire l'effort nécessaire à la création de données annotées pour l'entraînement, grâce à l'exploitation de corpus parallèles, de corpus faiblement annotés ou encore de connaissances fournies par des ressources existantes, comme l'UMLS (Lindberg *et al.*, 1993).

2.1 Méthodes non supervisées

Dans un travail, les chercheurs utilisent un corpus parallèle anglais-allemand (*Spinger Corpus of Medical Abstracts*) pour effectuer la désambiguïsation de termes médicaux (Widdows *et al.*, 2003). L'avantage de ce type de corpus est qu'un mot ambigu dans une langue ne l'est pas toujours dans

une autre langue : la mise en parallèle de ces deux langues permettrait donc de désambiguïser les occurrences dans l'une des langues. L'annotation est effectuée automatiquement en utilisant les CUI [1] de l'UMLS. Ainsi, pour un terme ambigu dans le résumé en anglais, les auteurs cherchent sa traduction dans le résumé correspondant en allemand. Les traductions sont gardées uniquement si (1) seulement un CUI est assigné à n'importe quel terme du résumé en allemand et (2) au moins un des termes, auquel le CUI est assigné dans le résumé en allemand, n'est pas ambigu. De cette manière, la désambiguïsation est effectuée pour les termes ambigus en anglais et les termes ambigus en allemand, tout en utilisant le même corpus de textes parallèles. Pour les termes en anglais, cette méthode donne une précision de 81 % et un rappel de 18 %. Pour la désambiguïsation des termes en allemand, elle donne une précision de 66 % et un rappel de 22 %.

Une autre méthode proposée par le même groupe de chercheurs (Widdows *et al.*, 2003) repose sur les collocations. En effet, il a été observé que les mots et termes ambigus ont tendance à avoir plusieurs collocations, parmi lesquelles une collocation donnée peut correspondre à un sens donné (Yarowsky, 1993). Pour rendre cette propriété totalement non supervisée, les auteurs utilisent également les CUI de l'UMLS, associés alors aux sens (un CUI est supposé correspondre à un sens), pour chaque mot ambigu afin de déterminer ses collocations. Cette méthode ne fournit pas un très bon rappel (3 % sur le corpus en anglais et 1 % sur le corpus en allemand), mais elle donne une précision assez élevée : 79 % pour les termes en anglais et 82 % pour les termes en allemand.

Enfin, une dernière méthode proposée par ce groupe de chercheurs (Widdows *et al.*, 2003) consiste en l'utilisation de termes qui sont liés par des relations conceptuelles contenues dans les tables MRREL et MRCXT de l'UMLS. Ainsi, pour chaque sens d'un mot ambigu w, la méthode cherche les termes qui sont liés à ce sens (également représenté par un CUI) dans les fichiers MRREL et MRCXT. Ensuite, pour chaque occurrence du terme ambigu w, le contexte est identifié et chaque mot du contexte est recherché dans les fichiers liés à chaque sens de w : si le mot du contexte fait partie du sens en particulier alors le score de ce sens est incrémenté. À la fin du processus, le sens qui a le score le plus haut est considéré comme celui qui correspond au sens du mot ambigu w. Cette méthode montre une précision entre 71 et 74 % pour la désambiguïsation de termes en anglais, et une précision entre 77 et 79 % pour la désambiguïsation de termes en allemand. Concernant le rappel, il est entre 32 et 49 % pour les termes en anglais, et entre 31 et 58 % pour les termes en allemand.

2.2 Méthodes supervisées

Un travail, effectué sur les données de la langue médicale exploite une méthode en deux étapes (Liu & Lussier, 2001). Lors de la première étape, pour un terme w ambigu, les auteurs définissent automatiquement un corpus étiqueté sémantiquement grâce à trois ressources (*UMLS Metathesaurus*, *MEDLINE* (NLM, 2015) et *Clinical Data Repository*). La seconde étape correspond alors à la construction d'un classifieur dédié à la désambiguïsation avec plusieurs algorithmes d'apprentissage supervisé (*Naive Bayes*, *Decision List* et *Exemplar-based*). Il s'agit donc d'une méthode mixte. Cette méthode méthode montre une *accuracy* entre 75 et 99 %.

D'autres chercheurs utilisent également les informations sur les CUI de l'UMLS (McInnes *et al.*, 2007). La méthode proposée consiste à obtenir les CUI des termes qui se trouvent dans la même fenêtre contextuelle que le terme ambigu. Dans ce travail, la fenêtre peut correspondre à la phrase dans laquelle se trouve le mot ambigu ou même au résumé complet. Lorsque les CUI contextuels sont

1. *Concept Unique Identifier* : il s'agit d'un code unique qui représente un concept (ensemble de termes sémantiquement équivalents) défini dans le Metathesaurus de l'UMLS.

définis, ils sont assignés manuellement au terme ambigu. Ensuite, le système calcule la fréquence qui correspond au nombre de fois où ce CUI apparaît dans le même contexte que le terme ambigu. Les chercheurs utilisent l'algorithme d'apprentissage Naive Bayes en validation croisée, tel qu'implémenté dans la plateforme WEKA (Witten & Frank, 2005).

Une autre approche proposée (Mihăltz, 2005) détermine d'abord, pour chaque terme ambigu, les informations syntaxiques d'autres mots se trouvant dans la même phrase que ce terme ambigu. Ensuite, le système détermine le domaine sémantique, ou le sujet, du paragraphe dans lequel se trouve le terme ambigu, ce qui permet de définir le sens de ce terme. Ce travail, effectué sur les données en hongrois et en anglais, montre une précision de 76,39 % pour l'anglais et de 84,2 % pour le hongrois.

Un autre chercheur (Pedersen, 2001) assigne un sens à un terme ambigu en exploitant les bigrammes qui se trouvent dans le contexte de ce terme. Différents algorithmes de WEKA sont alors utilisés (les arbre de décision *J48*, *Decision Stump* et *Naive Bayes*).

Une autre méthode exploite le *bilingual bootstrapping* (Li & Li, 2004). Cette méthode consiste à utiliser un petit volume de données annotées et un grand volume de données non annotées dans deux langues (langue source et langue cible). La méthode construit des classifieurs dans ces deux langues en parallèle et renforce la performance des classifieurs, d'une part en classifiant les données de chaque langue et, d'autre part, en échangeant des informations sur les données classifiées dans les deux langues.

Enfin, dans un dernier travail que nous voulons présenter, les chercheurs travaillent spécifiquement sur la désambiguïsation des abréviations du domaine médical (Stevenson *et al.*, 2009). Ils utilisent plusieurs algorithmes d'apprentissage supervisé (*Vector Space Model*, *Naive Bayes* et *Support Vector Machines*). Plusieurs descripteurs et paramètres sont pris en compte pour créer les modèles de désambiguïsation :

— les collocations avec les bi-grammes et tri-grammes (de lemmes, de formes et de parties du discours), de même que les couples forme/lemme se trouvant dans la même phrase que l'abréviation ambiguë,
— les CUI, selon l'approche de (McInnes *et al.*, 2007),
— l'utilisation de termes de Medical Subject Headings (MeSH), qui sont exploités pour indexer les documents médicaux dans MEDLINE. Ces termes sont associés manuellement aux différents résumés. Ainsi, sont utilisés comme descripteurs les termes MeSH qui sont associés aux résumés dans lesquels se trouvent les termes ambigus.

Cette méthode montre une performance entre 0,954 et 0,990.

3 Méthodologie

Notre méthode est une méthode supervisée qui s'appuie sur cinq classifieurs tels qu'implémentés dans la librairie ScikitLearn (Pedregosa *et al.*, 2011). Nous décrivons les données exploitées, les paramètres de la méthode et les principes de l'évaluation.

Les abréviations, le corpus d'entraînement et le corpus de test sont issus du corpus CLEAR (Grabar & Cardon, 2018). Il s'agit d'un corpus composé de trois sous-corpus de textes comparables : un sous-corpus de notices de médicaments, un sous-corpus de résumés de revues systématiques et un sous-corpus d'articles d'encyclopédies en ligne gratuites. Ce corpus contient en effet de nombreuses abréviations du domaine médical. Nous avons également constaté que plusieurs de ces abréviations

sont ambiguës, comme les exemples en (2) présentés dans la section 1.

3.1 Ensemble d'abréviations pour la désambiguïsation

Parmi les 1 638 abréviations détectées dans ce corpus, 138 sont ambiguës. Nous travaillons donc avec ces 138 abréviations. Chaque abréviation est associée avec l'ensemble de ses développements connus.

Sur les 138 abréviations ambiguës, 34 ne sont pas exploitables parce que le nombre d'occurrences est trop faible :

— 11 abréviations n'ont qu'une seule occurrence d'un des développements possibles,
— 7 abréviations n'ont que deux occurrences d'un des développements possibles,
— 16 abréviations n'ont que 3 à 5 occurrences d'un des développements possibles.

Les expériences sont donc effectuées avec 104 abréviations.

Les abréviations ont des niveaux d'ambiguïté différents avec 2 à 7 sens possibles. Le tableau 1 indique le nombre de développements possibles pour ces 104 abréviations. Nous pouvons voir que les abréviations avec 2 sens sont les plus fréquentes mais qu'il n'est pas rare d'avoir des abréviations avec plus de 2 sens.

TABLE 1: Ambiguïté des abréviations : nombre de sens ou de développements possibles.

	Nombre	Exemple
2 développements	72	*VC (ventilation conventionnelle, volume courant)*
3 développements	18	*TRC (taux rémission complète, taux réponse complète, temps recoloration cutanee)*
4 développements	10	*TSA (traitement suppression androgénique, traumatisme sonore aigu, travailleur santé autochtone, Trouble Spectre Autisme)*
>4 développements	4	*RC (réadaptation cardiaque, régime cétogène, rémunération conditionnelle, reponse conditionnelle, rythme cardiaque, rapport des cotes, rémission complète)*

3.2 Données de référence

Les documents du corpus CLEAR sont annotés par l'étiqueteur et l'analyseur syntaxique Cordial (Laurent *et al.*, 2009). Les phrases qui comportent les abréviations ambiguës sont exploitées. Nous avons construit deux corpus de référence :

— *Corpus d'entraînement*. Le corpus d'entraînement contient les phrases avec les formes étendues des abréviations ambiguës. Au total, le corpus contient 174 099 phrases. Ces données de référence sont créées automatiquement car les formes étendues des abréviations ne sont pas ambiguës. Cela représente une moyenne de 1 674 phrases par abréviation. Le minimum d'exemples (n=11) est observé avec l'abréviation *TRA (techniques de reproduction assistée, traitement restaurateur atraumatique, traitement de restauration atraumatique)*. Le maximum d'exemples (n=25 885) est observé avec l'abréviation *PA (phosphatase alkaline, pression*

artérielle, Pseudomonas aeruginosa). Concernant les sens des abréviations, nous avons une moyenne de 662 phrases par sens, avec un minimum d'une seule occurrence (19 formes étendues concernées) et un maximum de 25 455 occurrences pour la forme étendue *pression artérielle*. Nous pouvons voir que, selon les abréviations et les sens, le nombre d'exemples disponibles est plus ou moins élevé. Les données d'entraînement ne sont donc pas équilibrées. Les formes étendues des abréviations sont lemmatisées en même temps que les phrases. Ainsi, lorsqu'il existe plus d'une forme flexionnelle possible (comme *groupes hospitaliers* et *groupe hospitalier* pour *GH*) elles sont groupées ensemble. En revanche, les formes dérivationnelles (comme *traitement restaurateur atraumatique* et *traitement de restauration atraumatique* pour *TRA*) ne sont pas groupées ensemble ;

— *Corpus de test*. Le corpus de test comporte les phrases avec les abréviations ambiguës. Au total, le corpus contient 1 665 phrases. 92 des 104 abréviations ambiguës sont présentes dans le corpus de test mais avec des contextes différents. Ces données de référence sont créées manuellement. Pour chaque phrase, la décision sur le bon sens (forme développée) d'une abréviation est prise grâce au contexte de la phrase et, lorsque cela n'est pas suffisant, nous consultons le document duquel est extraite la phrase.

Ainsi, dans le corpus d'entraînement, nous avons les phrases avec les formes étendues (exemple en (3)), alors que, dans le corpus de test, nous avons les phrases avec les abréviations (exemple en (4)).

(3) *Déterminer l'efficacité des interventions comportementales pour traiter une dysménorrhée primaire ou secondaire les unes par rapport aux autres, par rapport à un placebo, à l'absence de traitement ou à des traitements médicaux conventionnels, par exemple les anti-inflammatoires non stéroïdiens (AINS).*

(4) *En raison du caractère limité des résultats soutenant le recours à la neurectomie antéro-sacrée pour la prise en charge de la DP, les risques doivent être rigoureusement mis en balance avec les avantages attendus.*

Pour la création de descripteurs, nous n'utilisons pas de connaissances externes mais uniquement les informations contextuelles contenues dans les phrases du corpus. Ainsi, pour chaque sens de chaque abréviation ambiguë, nous cherchons ses contextes dans une fenêtre de cinq mots à gauche et de cinq mots à droite. Lorsque le sens de l'abréviation est suivi de l'abréviation elle-même entre parenthèses, nous ne prenons pas en compte les parenthèses et l'abréviation se trouvant entre parenthèses dans le contexte de cinq mots à droite, mais passons directement au premier mot se trouvant après la parenthèse. Ainsi, dans la phrase *la différence moyenne entre les groupes était de -0,18 LogMAR (intervalle de confiance (IC) à 95 % statistiquement significatif de -0,32 à -0,04).*, les mots du contexte de droite seront *à*, *95 %*, *statistiquement*, *significatif* et *de*. Les mots lexicaux et les mots grammaticaux sont pris en compte. Le contexte est représenté par les lemmes et par les étiquettes syntaxiques des co-occurrences des abréviations. De plus, la position de chaque co-occurrence est également retenue. Par exemple, pour illustrer cette transformation, nous utilisons la phrase en (3) pour la forme étendue *dysménorrhée primaire* de l'abréviation *DP*. Cette phrase produit les descripteurs suivants pour les contextes gauche et droit : *posi1-gauche_un*, *posi1-gauche_DETIFS*, *posi2-gauche_traiter*, *posi2-gauche_VINF*, *posi3-gauche_pour*, *posi3-gauche_PREP*, *posi4-gauche_comportemental*, *posi4-gauche_ADJFP*, *posi5-gauche_intervention*, *posi5-gauche_NCFP*, *posi1-droite_ou*, *posi1-gauche_COO*, *posi2-droite_secondaire*, *posi2-gauche_ADJSIG*, *posi3-droite_le*, *posi3-gauche_DETDPIG*, *posi4-droite_un*, *posi4-gauche_PIFP*, *posi5-droite_par_rapport_aux*, *posi5-gauche_PREP*. Ensuite, pour l'ensemble de descripteurs, nous calculons si un descripteur donné se trouve dans le contexte concerné

d'une occurrence d'une abréviation donnée. Si c'est le cas, la valeur du descripteur est 1 et si non sa valeur est 0. Le même calcul de descripteurs est effectué sur les phrases du corpus de test.

3.3 Algorithmes pour l'apprentissage supervisé

Nous exploitons les algorithmes implémentés dans la librairie ScikitLearn (Pedregosa *et al.*, 2011) destinée à l'apprentissage supervisé et non supervisé. Nous avons choisi d'utiliser cinq classifieurs pour un apprentissage supervisé :

— SVM Linear et SVM RBF (Platt, 1998). SVM est un algorithme d'apprentissage supervisé qui peut être utilisé à la fois pour la classification et pour la régression. Ici, nous l'utilisons pour la classification. Il s'agit d'un algorithme qui cherche un hyperplan pour mieux séparer les paramètres des classes. Nous utilisons deux noyaux : linéaire et Gaussien (RBF) ;

— Decision Tree (Quinlan, 1993). Un arbre de décision est représenté sous la forme d'un arbre, où une décision possible est située à chaque embranchement. Elle est atteinte ou non en fonction des choix effectués à chaque étape de l'arbre ;

— MultiLayer Perceptron (Rosenblatt, 1961). Un perceptron multicouche est composé de plusieurs couches dans lesquelles circule une information. Les dernières couches représentent la sortie du système ;

— RandomForest (Breiman, 2001). Les forêts d'arbres décisionnels fonctionnent grâce à un apprentissage effectué sur différents arbres de décision entraînés sur des sous-ensembles de données.

L'ensemble de descripteurs (contextes gauche et droit) est exploité avec les algorithmes. Les algorithmes doivent prédire le sens d'une abréviation ambiguë en fonction du contexte où elle apparaît.

3.4 Évaluation

L'entraînement est effectué sur le corpus d'entraînement et le test est effectué avec les 1 665 phrases du corpus de test. Sur le corpus d'entraînement, pour chaque abréviation, nous effectuons une validation croisée à 10 plis. En fonction du nombre de sens, les modèles sont bi-classes ou multi-classes. Les modèles créés sur ce corpus d'entraînement sont appliqués et testés sur le corpus de test. Sur le corpus de test, nous gardons uniquement la première prédiction, dont la probabilité est la plus grande, pour chaque occurrence d'abréviation et la comparons avec les données de référence. Nous calculons les mesures d'évaluation classiques (Sebastiani, 2002) : Précision, Rappel et F-mesure dans leurs versions micro. Nous calculons également la moyenne de ces mesures pour chaque algorithme.

Notre *baseline* correspond à la catégorisation des sens dans la catégorie majoritaire. Les résultats de la baseline sont également évalués en termes de Précision, Rappel et F-mesure.

4 Résultats

Le tableau 2 indique les résultats obtenus avec une validation croisée à 10 plis sur le corpus d'entraînement. Nous pouvons constater que ces résultats sont assez élevés et que l'algorithme Multi-Layer Perceptron (MLP) obtient de meilleurs résultats avec une F-mesure de 0,888. Les valeurs de Précision et de Rappel sont équilibrées pour les différents algorithmes testés. Ces résultats en validation croisée

sont comparables avec les résultats de l'état de l'art (Liu & Lussier, 2001; Mihăltz, 2005). La dernière colonne du tableau indique les performances obtenues avec la *baseline*. Nous voyons que tous les algorithmes exploités montrent des résultats supérieurs à la *baseline*.

TABLE 2: Résultats obtenus sur le corpus l'entraînement par chaque algorithme en validation croisée.

Mesure	SVM Linear	SVM RBF	Decision Tree	MLP	RandomForest	Baseline
Rappel	0,885	0,878	0,897	0,905	0,887	0,822
Précision	0,880	0,849	0,887	0,892	0,871	0,822
F-mesure	0,877	0,856	0,888	0,895	0,873	0,822

Les résultats de désambiguïsation des abréviations dans le corpus de test, obtenus avec différents algorithmes testés, sont présentés dans le tableau 3. Nous voyons que Decision Tree présente les meilleurs résultats avec 0,773 de F-mesure et les valeurs de Précision et de Rappel assez équivalentes. Les autres algorithmes montrent un Rappel beaucoup plus bas, ce qui diminue leurs performances globales. Nous remarquons que les résultats de la *baseline* sont supérieurs aux résultats fournis par les algorithmes.

TABLE 3: Résultats de la désambiguïsation dans le corpus de test.

Mesure	SVM Linear	SVM RBF	Decision Tree	MLP	RandomForest	Baseline
Rappel	0,402	0,398	0,788	0,424	0,402	0,822
Précision	0,797	0,755	0,759	0,763	0,728	0,822
F-mesure	0,534	0,524	0,773	0,545	0,518	0,822

Le tableau 4 indique le nombre d'abréviations qui ont été correctement traitées et désambiguïsées par les différents algorithmes. Nous pouvons voir que Decision Tree arrive à traiter correctement le plus grand nombre d'occurrences (547).

TABLE 4: Nombre total d'occurrences des abréviations classifiées correctement dans le corpus de test.

SVM Linear	SVM RBF	Decision Tree	MLP	RandomForest
441	516	547	523	492

Parmi les abréviations qui sont correctement désambiguïsées avec 100 % de prédictions correctes, nous avons par exemple *DIU (diplôme inter universitaire, dispositif intra utérin)* et *GH (groupe hospitalier, growth hormone)*. Nous voyons deux raisons principales à cela : (1) les sens de ces deux abréviations ont des sémantiques très éloignées et donc des contextes très différents et (2) ces abréviations ont de nombreux exemples dans les données d'entraînement. Pour ces deux raisons, leur désambiguïsation est facilitée. 18 autres abréviations sont dans ce cas également. De plus, leur désambiguïsation s'avère aisée pour tous les algorithmes testés. Plusieurs autres abréviations montrent des performances variées selon les algorithmes, dont les valeurs peuvent aller de 0 à 100 %. Finalement, pour plusieurs abréviations (comme *APS, ASA, HE* et *TRA*), nous n'obtenons

malheureusement pas de bons résultats. Il nous semble que la raison principale est que les exemples pour ces abréviations sont insuffisants voire même absents dans le corpus d'entraînement. Il est donc nécessaire de compléter le corpus d'entraînement avec d'autres occurrences de formes étendues.

Globalement, même si les résultats obtenus sur le corpus de test sont inférieurs à ceux obtenus sur le corpus d'entraînement, ils restent comparables avec les résultats obtenus par d'autres chercheurs dans les travaux existants. Nous pensons que le rappel baisse autant entre la validation croisée dans le corpus d'entraînement et le corpus de test parce que les contextes dans lesquels se trouvent les abréviation et leurs versions étendues sont différents. Ces contextes permettent cependant d'effectuer la désambiguïsation de manière très efficace. Étant donné les résultats obtenus, l'algorithme Decision Tree semble être le plus approprié pour effectuer la tâche de désambiguïsation. Un de ses points forts est de garder les valeurs équilibrées pour la Précision et le Rappel. Nos résultats indiquent cependant qu'il est nécessaire d'apporter plusieurs améliorations à ce travail. L'amélioration la plus importante consiste à enrichir le corpus d'entraînement avec d'autres exemples pour les abréviations et les sens qui n'en disposent pas suffisamment actuellement.

5 Conclusion et discussion

Nous avons présenté notre travail sur la désambiguïsation d'abréviations du domaine médical. Après étude des différents moyens pour effectuer la désambiguisation, nous avons proposé d'utiliser une approche par catégorisation supervisée. L'exploitation de ressources sémantiques, comme la terminologie MESHS utilisée dans un travail existant (Stevenson *et al.*, 2009), reste une perspective pour les travaux futurs. De plus, nous disposons d'un nombre assez importants d'exemples. Ainsi, l'entraînement est effectué sur des phrases, obtenues à partir du corpus CLEAR, qui contiennent les formes étendues, et donc non ambiguës, des abréviations. Le test est effectué sur un corpus construit manuellement, où les bons sens des abréviations ont été définis selon leurs contextes phrastiques ou, si nécessaire, la consultation du document d'origine. L'utilisation de ces deux types d'évaluation (validation croisée et corpus de test) montre que, bien que les résultats en validation croisée soient prometteurs, ce n'est pas forcément le cas lorsque le corpus de test est utilisé. L'approche d'apprentissage supervisé (algorithme Decision Tree) montre actuellement une F-mesure moyenne de 0,888 sur le corpus d'entraînement en validation croisée et 0,773 sur le corpus de test. Les résultats montrés par la *baseline*, où l'on assigne les sens à la catégorie majoritaire, sont supérieurs dans le corpus de test. Nous pensons que le déséquilibre entre les catégories devrait être réduit pour disposer de plus d'exemples et obtenir de meilleurs résultats.

Nous avons ainsi plusieurs pistes pour le travail à venir :

— compléter le corpus d'entraînement avec d'autres exemples, notamment tirés de documents médicaux autre que le corpus CLEAR, ce qui permettraient d'améliorer les résultats pour plusieurs abréviations actuellement sous-représentées,
— compléter l'ensemble d'abréviations avec d'autres développements possibles, ce qui peut conduire à l'augmentation de l'ambiguïté de certaines abréviations (plus de sens connus) et à l'augmentation du nombre d'abréviations ambiguës,
— mettre à jour les modèles et créer de nouveaux modèles pour la désambiguïsation d'abréviations.

Finalement, ce système de désambiguïsation sera intégré dans un système plus global dédié à la simplification de textes techniques du domaine médical.

Remerciements

La présente publication s'inscrit dans le projet *CLEAR (Communication, Literacy, Education, Accessibility, Readability)* financé par l'ANR sous la référence ANR-17-CE19-0016-01. Nous remercions les relecteurs pour leurs remarques constructives.

Références

BIKEL D. M. (2000). A statistical model for parsing and word-sense disambiguation. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora : Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, p. 155–163, USA : Association for Computational Linguistics. DOI : 10.3115/1117794.1117814.

BREIMAN L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.

BROWN P. F., DELLA PIETRA S. A., DELLA PIETRA V. J. & MERCER R. L. (1991). A statistical approach to sense disambiguation in machine translation. In *Speech and Natural Language : Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

GRABAR N. & CARDON R. (2018). Clear – simple corpus for medical French. In *Workshop on Automatic Text Adaption (ATA)*, p. 1–11.

IDE N. & VÉRONIS J. (1998). Introduction to the special issue on word sense disambiguation : The state of the art. *Computational Linguistics*, **24**(1), 1–40.

KROVETZ R. (2002). On the importance of word sense disambiguation for information retrieval.

LAURENT D., NÈGRE S. & SÉGUÉLA P. (2009). L'analyseur syntaxique Cordial dans Passage. In *Traitement Automatique des Langues Naturelles (TALN)*.

LI H. & LI C. (2004). Word translation disambiguation using bilingual bootstrapping. *Computational Linguistics*, **30**(1), 1–22. DOI : 10.1162/089120104773633367.

LIM L. T. & TANG E. K. (2004). Building an ontology-based multilingual lexicon for word sense disambiguation in machine translation.

LINDBERG D., HUMPHREYS B. & MCCRAY A. (1993). The Unified Medical Language System. *Methods Inf Med*, **32**(4), 281–291.

LIU H. & LUSSIER Y. (2001). Disambiguating ambiguous biomedical terms in biomedical narrative text : An unsupervised method. *Journal of Biomedical Informatics*, **34**, 249–261. DOI : 10.1006/jbin.2001.1023.

MARVIN R. & KOEHN P. (2018). Exploring word sense disambiguation abilities of neural machine translation systems (non-archival extended abstract). In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1 : Research Papers)*, p. 125–131, Boston, MA : Association for Machine Translation in the Americas.

MCINNES B. T., PEDERSEN T. & CARLIS J. (2007). Using umls concept unique identifiers (cuis) for word sense disambiguation in the biomedical domain. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, **2007**, 533–537.

MIHÁLTZ M. (2005). Towards a hybrid approach to word-sense disambiguation in machine translation.

NLM (2015). *Medline : medical literature on-line*. National Library of Medicine, Bethesda, Maryland. www.ncbi.nlm.nih.gov/sites/entrez.

OCDE (2015). *Guide de style de l'OCDE Troisième édition : Troisième édition*. OECD Publishing.

PARAMESWARAPPA S. & NARAYANA V. (2011). Article : Kannada word sense disambiguation for machine translation. *International Journal of Computer Applications*, **34**(10), 1–8. Full text available.

PEDERSEN T. (2001). A decision tree of bigrams is an accurate predictor of word sense. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M. & DUCHESNAY E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

PLATT J. C. (1998). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning* : MIT Press.

QUINLAN J. (1993). *C4.5 Programs for Machine Learning*. San Mateo, CA : Morgan Kaufmann.

ROSENBLATT F. (1961). *Principles of Neurodynamics : Perceptrons and the Theory of Brain Mechanisms*. Washington DC : Spartan Books.

RUEL J., KASSI B. M. A. C. & L. M.-M. S. (2011). *Guide de rédaction pour une information accessible*. Gatineau : Pavillon du Parc.

SEBASTIANI F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**(1), 1–47.

SPECIA L. (2005). A hybrid model for word sense disambiguation in english-portuguese machine translation. In *IN PROCEEDINGS OF THE 8TH RESEARCH COLLOQUIUM OF THE UK SPECIAL-INTEREST GROUP IN COMPUTATIONAL LINGUISTICS*, p. 71–78.

STEVENSON M., GUO Y., ALAMRI A. & GAIZAUSKAS R. (2009). Disambiguation of biomedical abbreviations. In *Proceedings of the BioNLP 2009 Workshop*, p. 71–79, Boulder, Colorado : Association for Computational Linguistics.

STOKOE C., OAKES M. P. & TAIT J. (2003). Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, p. 159–166, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/860435.860466.

STOKOE C. & TAIT J. (2002). Trec 2002 web track "automated word sense disambiguation for internet information retrieval".

TANG G., SENNRICH R. & NIVRE J. (2018). An analysis of attention mechanisms : The case of word sense disambiguation in neural machine translation. *CoRR*, **abs/1810.07595**.

UNAPEI (2019). *L'information pour tous*. UNAPEI.

VICKREY D., BIEWALD L., TEYSSIER M. & KOLLER D. (2005). Word-sense disambiguation for machine translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, p. 771–778, USA : Association for Computational Linguistics. DOI : 10.3115/1220575.1220672.

WHALEY J. M. (1999). An application of word sense disambiguation to information retrieval.

WIDDOWS D., PETERS S., CEDERBERG S., CHAN C.-K., STEFFEN D. & BUITELAAR P. (2003). Unsupervised monolingual and bilingual word-sense disambiguation of medical documents using umls. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine - Volume 13*, BioMed '03, p. 9–16, USA : Association for Computational Linguistics. DOI : 10.3115/1118958.1118960.

WITTEN I. & FRANK E. (2005). *Data mining : Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

YAROWSKY D. (1993). One sense per collocation. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, p. 266–271, USA : Association for Computational Linguistics. DOI : 10.3115/1075671.1075731.

ZHONG Z. & NG H. T. (2012). Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers - Volume 1*, ACL '12, p. 273–282, USA : Association for Computational Linguistics.

ZHOU X. & HAN H. (2005). Survey of word sense disambiguation approaches.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Apprentissage de plongements de mots sur des corpus en langue de spécialité : une étude d'impact

Valentin Pelloin Thibault Prouteau
LIUM, Avenue Laennec, 72085 LE MANS, France
valentin.pelloin.etu@univ-lemans.fr,
thibault.prouteau.etu@univ-lemans.fr

RÉSUMÉ

Les méthodes d'apprentissage de plongements lexicaux constituent désormais l'état de l'art pour la représentation du vocabulaire et des documents sous forme de vecteurs dans de nombreuses tâches de Traitement Automatique du Langage Naturel (TALN). Dans ce travail, nous considérons l'apprentissage et l'usage de plongements lexicaux dans le cadre de corpus en langue de spécialité de petite taille. En particulier, nous souhaitons savoir si dans ce cadre, il est préférable d'utiliser des plongements préappris sur des corpus très volumineux tels Wikipédia ou bien s'il est préférable d'apprendre des plongements sur ces corpus en langue de spécialité. Pour répondre à cette question, nous considérons deux corpus en langue de spécialité : OHSUMED issu du domaine médical, et un corpus de documentation technique, propriété de SNCF. Après avoir introduit ces corpus et évalué leur spécificité, nous définissons une tâche de classification. Pour cette tâche, nous choisissons d'utiliser en entrée d'un classifieur neuronal des représentations des documents qui sont soit basées sur des plongements appris sur les corpus de spécialité, soit sur des plongements appris sur Wikipédia. Notre analyse montre que les plongements appris sur Wikipédia fournissent de très bons résultats. Ceux-ci peuvent être utilisés comme une référence fiable, même si dans le cas d'OHSUMED, il vaut mieux apprendre des plongements sur ce même corpus. La discussion des résultats se fait en interrogeant les spécificités des deux corpus, mais ne permet pas d'établir clairement dans quels cas apprendre des plongements spécifiques au corpus.

ABSTRACT

Learning word embeddings on domain specific corpora : an impact study

Word embedding approaches are state of the art in Natural Language Processing (NLP). In this work, we focus on learning word embeddings for small domain-specific corpora. In particular, we would like to know whether word embeddings learnt over large corpora such as Wikipedia perform better than word embeddings learnt on domain specific corpora. In order to answer this question, we consider two corpora : OHSUMED from the medical field, and SNCF, a technical documentation corpus. After presenting the corpora and evaluating their specificity, we introduce a classification task. We use word embeddings learnt on domain-specific corpora or Wikipedia as input for this task. Our analysis demonstrates that word embeddings learnt on Wikipedia achieve excellent results, even though, in the case of OHSUMED, domain specific word embeddings perform better.

MOTS-CLÉS : langue de spécialité, plongements de mots, catégorisation de documents.

KEYWORDS: domain specific, word embeddings, documents categorization.

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 164–178.
HAL : hal-02786198.

1 Introduction

Les approches contemporaines en traitement automatique des langues privilégient une représentation des mots au travers de *plongements de mots* (ou *plongements lexicaux*) décrits notamment par Mikolov *et al.* (2013) et Bojanowski *et al.* (2017). Les plongements lexicaux sont des représentations vectorielles compactes des mots, qui encapsulent les liens sémantiques et syntaxiques présents dans les textes des corpus sur lesquels ils ont été appris. L'apprentissage de plongements de mots s'effectue le plus souvent sur un corpus comportant une grande quantité de documents en langue générale comme Wikipédia (Huang *et al.*, 2012; Liu *et al.*, 2015; Yamada *et al.*, 2018).

Dans notre cas, nous nous concentrons sur la langue de spécialité. La langue de spécialité est celle qui est utilisée pour rendre compte de connaissances sur un sujet ou une discipline spécialisée (Charnock, 1999; Van Der Yeught, 2016; Schmitt, 2002; Condamines, 1997; Chujo & Utiyama, 2006; Paltridge & Starfield, 2016). Par conséquent, le lexique employé est la plupart du temps imposé par la discipline abordée et le vocabulaire employé diffère donc de celui que l'on retrouve dans un corpus non spécialisé. Ainsi, il semble nécessaire, pour représenter efficacement le vocabulaire et les documents de tels corpus, d'apprendre les plongements lexicaux sur ces mêmes corpus. Cependant, la taille de ces corpus est très nettement inférieure à celle de corpus comme Wikipédia, ce qui peut nuire à l'apprentissage des plongements qui repose sur les cooccurrences du vocabulaire. Or, plus le nombre de cooccurrences est faible, moins l'algorithme dispose d'informations pour apprendre une représentation du lexique. De ce fait, on cherche à savoir quelle représentation adopter lorsque l'on traite des données en langue de spécialité : est-il préférable d'utiliser des plongements lexicaux appris sur une grande quantité de données non spécialisés ou sur une faible quantité de données spécialisées ?

Plusieurs études s'intéressent à l'apprentissage de plongements lexicaux sur un corpus spécialisé du domaine médical (El Boukkouri *et al.*, 2019; Lee *et al.*, 2020) en adaptant des modèles appris sur des données en langue générale au domaine médical. Wang *et al.* (2018) étudient l'apprentissage de plongements lexicaux sur des corpus médicaux et concluent que les plongements lexicaux appris sur un corpus spécialisé capturent mieux les liens sémantiques et syntaxiques que ceux appris sur un corpus non *spécialisé*. Cet article s'inscrit dans la continuité de ces travaux en proposant une étude sur deux corpus en langue de spécialité : OHSUMED, un corpus issu du domaine médical, et SNCF, un corpus *ad hoc* de documents techniques. À la différence des approches proposées par (El Boukkouri *et al.*, 2019; Lee *et al.*, 2020) nous décidons de comparer les performances des modèles appris sur des données en langue générale et spécialisée ainsi qu'une spécialisation aussi bien à une tâche de classification de document qu'au domaine spécialisé.

Dans un premier temps, nous tentons d'estimer la spécificité de ces deux corpus automatiquement, à travers plusieurs indicateurs extraits de la littérature et décrits Section 2.1. Ensuite, Section 2.2, nous détaillons le protocole d'apprentissage des plongements de mots sur les corpus spécialisés OHSUMED et SNCF, et sur les corpus non spécialisés. Dans cette même section, nous présentons le protocole mis en oeuvre pour évaluer les performances des différents plongements appris. Ce protocole repose sur une tâche de classification en classes multiples par un réseau neuronal convolutif. Puis, nous décrivons les corpus OHSUMED et SNCF Section 3. Enfin, nous détaillons Section 4 les résultats avant de conclure et de discuter des perspectives de ce travail en cours.

2 Méthodologie

L'objectif de ce travail est de savoir, si utiliser des plongements de mots préappris sur des corpus très volumineux tels que Wikipédia est préférable à l'utilisation de plongements lexicaux appris sur nos

corpus en langue de spécialité.

Avant de détailler nos expérimentations sur les plongements lexicaux dans le cadre d'une utilisation sur un corpus spécialisé, nous commençons par définir ce qu'est un corpus spécialisé et présentons quelques indicateurs de la spécialité d'un corpus.

2.1 Estimer la spécificité

Le contenu des textes d'un corpus spécialisé correspond à un domaine particulier et s'adresse aux initiés dudit domaine. De plus, le domaine impose le vocabulaire et le style employé (Charnock, 1999; Van Der Yeught, 2016).

Analyse du destinataire. Les corpus spécialisés ont la particularité d'être à destination d'un groupe limité d'individus. Dans le cas du corpus SNCF les documents sont destinés aux agents de l'entreprise. Concernant le corpus OHSUMED, les documents sont des *abstracts* d'articles scientifiques sur des maladies, ceux-ci sont destinés à des médecins et chercheurs en médecine.
Cette observation faite, on peut s'interroger sur le niveau d'expertise de la cible et son impact sur le lexique employé. Les documents à destination d'un public expert contiennent-ils plus de termes apparentés au jargon que ceux destinés à des néophytes ? Nous présentons ci-après plusieurs indicateurs sur le lexique.

Analyse du lexique. D'après Cressot & James (1996), le style d'un corpus peut se résumer aux choix réalisés par les auteurs lors de l'écriture des documents du corpus, ainsi, la lexicologie, la grammaire et la syntaxe peuvent entre autres permettre de décrire ce style. Nous étudions le style des documents en choisissant une approche lexicale restreinte au vocabulaire employé et à la taille des phrases.

Nous calculons pour un corpus *spécialisé* (resp. non *spécialisé*) la couverture de son vocabulaire par un dictionnaire en langue commune (Éq. 1 et 2). Cela constitue un indicateur permettant d'évaluer à quel point le vocabulaire d'un corpus supposé *spécialisé* est différent du vocabulaire d'un corpus non *spécialisé*. Cette méthode est un premier estimateur facile à calculer. Néanmoins, il ne prend pas en compte la terminologie employée pouvant avoir un sens ou une connotation différente dans la langue de spécialité (Cabré, 2002).

Pour calculer la *couverture*, nous nous dotons d'un dictionnaire de mots pour la langue commune. Dans notre cas, nous utilisons les livres français de la bibliothèque du projet Gutenberg [1], composée de 60.000 livres numérisés. Un dictionnaire a été construit à partir de ces textes par Pythoud (1998). Concernant le vocabulaire anglais, nous utilisons le vocabulaire fourni par le modèle de langage EN_CORE_WEB_LG de Honnibal & Montani (2017) présent dans SPACY. La mesure de couverture peut se calculer à deux niveaux : soit en ne considérant qu'une seule occurrence de chaque mot (vocabulaire, Éq. 1), soit en considérant chaque occurrence d'un mot (Éq. 2). Le ratio de couverture pour un corpus C par rapport à un dictionnaire D est calculé comme suit :

1. https://www.gutenberg.org

$$\text{Couv}_{\text{voc}}(C|D) = \frac{|V_C \cap V_D|}{|V_C|} \quad (1) \qquad \text{Couv}_{\text{occ}}(C|D) = \frac{\sum_{w_i \in V_C \cap V_D} \#_{w_i} C}{\#C} \quad (2)$$

avec V_X le vocabulaire du corpus X, $\#X$ le nombre d'unités lexicales du corpus X et $\#_m X$ le nombre d'occurrences de l'unité lexicale m dans le corpus X.

Cet indicateur permet de comparer deux corpus selon leur couverture lexicale. Ainsi, si la couverture au niveau vocabulaire ou au niveau occurrence d'un corpus C_A est plus élevée que celle d'un autre corpus C_B, cela indique que le corpus C_B fait appel à moins de mots non *spécialisés* que le corpus C_A.

Étudier la couverture du vocabulaire par un dictionnaire donne une première idée du lexique utilisé dans un corpus. Néanmoins, cette mesure demeure relativement naïve puisqu'elle est dépendante du dictionnaire utilisé. On propose donc d'étudier la diversité du lexique qui peut être estimée à partir du *type-token ratio* (TTR) : le ratio entre la taille du vocabulaire d'un document et le nombre d'unités lexicales. Cependant, si le nombre d'unités lexicales d'un corpus croît linéairement selon sa taille, la taille du vocabulaire suit une loi différente, la *Heap law* (Herdan, 1960). Le TTR ne permet ainsi pas de comparer deux corpus de taille différente. Pour corriger cela, l'indice *measure of textual lexical diversity* (MTLD) permet d'estimer la diversité lexicale des documents d'un corpus indépendamment de la taille de ces corpus (Mccarthy, 2006; McCarthy & Jarvis, 2010; Torruella & Capsada, 2013). Pour ce faire, le document est divisé en n échantillons d'unités lexicales consécutives de façon à obtenir un seuil de TTR fixé (usuellement $TTR = 0.72$). La valeur de MTLD est ensuite le ratio entre n et la taille $|D|$ du document. Plus l'indice de diversité lexicale est important, plus le lexique employé dans les documents est large.

Analyse de la lisibilité. La lisibilité est définie comme la facilité avec laquelle un lecteur peut comprendre un texte écrit. Son contenu (complexité du vocabulaire et de la syntaxe) et sa présentation permettent de la mesurer. Plusieurs mesures de lisibilité de textes existent. Ces métriques n'utilisent pas les mêmes informations pour qualifier la lisibilité d'une phrase. Nous avons donc décidé d'utiliser les métriques SMOG et ARI qui sont complémentaires : cela permet de mesurer l'influence de la longueur des mots et des phrases sur la lisibilité.

La mesure de lisibilité SMOG définie par GH (1969) pour caractériser la *readability*, c.-à-d. la difficulté qu'aura un lecteur à comprendre un document, fait l'hypothèse qu'il est possible d'estimer la complexité du document à partir du nombre de mots polysyllabiques. SMOG repose sur le nombre de mots contenant trois syllabes ou plus dans chaque échantillon (Éq. 3). Cette mesure correspond au nombre théorique d'années d'éducation nécessaire pour comprendre un texte. Nous calculons le score SMOG en sélectionnant aléatoirement 1000 échantillons de 30 phrases dans le corpus. Le résultat obtenu est la moyenne des scores SMOG pour ces 1000 échantillons.

$$\text{SMOG}(S) = 1.0430\sqrt{|\text{polysyllabes}_S|} + 3.1291 \quad (3)$$

avec $S \subset C$ un échantillon du corpus C pour lequel nous calculons le score SMOG, polysyllabes_S les mots avec 3 syllabes ou plus dans S.

D'une langue à l'autre, il existe des variations dans l'utilisation des mots polysyllabiques. Par exemple, Contreras *et al.* (1999) montrent que sur des corpus parallèles en français et en anglais, le score SMOG est toujours plus faible en anglais. À partir de ce constat, ils introduisent un cadre permettant de comparer le score SMOG obtenu sur deux corpus qui ne sont pas écrits dans la même langue. Ainsi, pour comparer un score SMOG obtenu sur un corpus français avec celui d'un corpus anglais, on applique la formule 4.

$$\text{SMOG}(En) = -1,35 + 0,17 \times \text{SMOG}(Fr) \tag{4}$$

La mesure de lisibilité ARI telle que définie par Senter & Smith (1967) fournit, comme SMOG, une estimation du nombre d'années d'éducation nécessaire pour comprendre un document. Cette métrique est calculée à partir du nombre de caractères, de mots, et de phrases pour un corpus C :

$$\text{ARI}(C) = 4.71 \left(\frac{\text{caractères}_C}{\text{mots}_C} \right) + 0.5 \left(\frac{\text{mots}_C}{\text{phrases}_C} \right) - 21.43 \tag{5}$$

Cette mesure a été construite pour des documents en anglais, les résultats pour les documents du corpus SNCF en français sont donc donnés à titre indicatif.

2.2 Apprentissage de plongements lexicaux et classification

Plongements lexicaux. Pour chaque corpus spécialisé (OHSUMED et SNCF) ou corpus non spécialisé, on apprend un ensemble de plongements de mots à l'aide de l'algorithme CBOW, tel que présenté par Mikolov *et al.* (2013). Chaque mot est représenté par un vecteur de dimension 300. Ces vecteurs sont construits à partir d'une fenêtre de contexte de 5 mots, avec un minimum de 1 ou 2 occurrences de chacun des mots du vocabulaire des corpus, selon la taille du corpus utilisé. Nous utilisons un échantillonnage négatif (*negative sampling*). Les autres paramètres sont ceux recommandés et initialisés par défaut dans l'implémentation Gensim (Řehůřek & Sojka, 2010). On désigne dans la suite de l'article les plongements appris sur les corpus non spécialisés comme plongements *génériques*.

Classification. Les documents des corpus OHSUMED et SNCF à notre disposition sont catégorisés en classes spécifiant le thème du document en question (voir Section 3). L'objectif est d'apprendre un classifieur capable de retrouver ces classes automatiquement. Nous choisissons un classifieur de type réseau de neurones convolutif particulièrement adapté à la classification de documents représentés par des plongements lexicaux. Nous faisons l'hypothèse que le réseau de neurones obtiendra de meilleures performances si les plongements en entrée sont les plus adaptés à la tâche. Ainsi, nous comparons Section 4.4 les résultats obtenus avec le classifieur en fonction des plongements utilisés : *spécialisés* ou *génériques*. L'architecture de ce réseau est détaillée ci-dessous. Les corpus sont séparés en corpus d'apprentissage (TRAIN), en corpus de développement afin d'adapter les hyperparamètres (DEV), et en corpus d'évaluation (TEST). Cette séparation est réalisée de façon stratifiée : les proportions de chacune des classes sont préservées dans les corpus de TRAIN, DEV et TEST.

Architecture des modèles. L'architecture des modèles de classification appris est similaire à celle introduite par Kim (2014) et décrite Figure 1. Chaque document est représenté par la concaténation verticale des plongements des mots du document. Cette représentation est de taille fixe, il s'agit des 3 000 premiers mots de chaque document. Si un document est plus long que cette limite il sera tronqué, s'il est plus court, alors la matrice sera complétée par des 0 (*padding*). Ensuite, l'architecture intègre deux blocs de convolution. Chacun de ces blocs contient : une convolution, une normalisation par lot (*batch normalisation*), une couche d'activation ReLU (Hahnloser *et al.*, 2000) et une couche de *max-pooling*. En sortie des convolutions, le réseau est constitué de trois couches linéaires (*feed forward*), suivies d'une fonction *softmax* afin d'obtenir une distribution de probabilité pour les classes possibles. La dernière couche linéaire comporte autant de neurones que le niveau de classification comporte de classes. Les deux premières comportent respectivement 250 et 100 neurones. Un abandon (*dropout*, (Hinton *et al.*, 2012)) de $p = 0.2$ est réalisé sur ces deux couches. Nous utilisons un optimiseur ADAM (Kingma & Ba, 2015). Afin de réduire le taux de surapprentissage, nous réalisons une régularisation des poids (*weight decay*, Weigend *et al.* (1991)), ainsi qu'un *early stopping* qui arrête l'apprentissage lorsque l'erreur sur le corpus de DEV augmente. Enfin, le pas d'apprentissage est réduit au fil du temps.

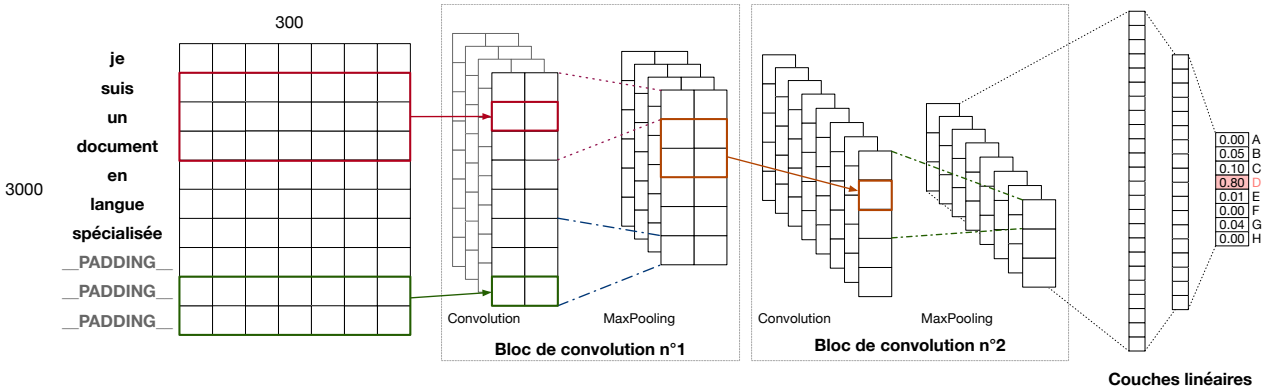


FIGURE 1 – Architecture de nos modèles de classification.

Lors de l'étape de *rétropropagation* (*backpropagation* en anglais) du gradient, deux choix sont possibles : faire la mise à jour des poids sur les couches linéaires et sur les blocs de convolution sans toucher aux plongements (non *trainable*), ou faire la mise à jour des poids jusqu'à la couche des plongements de mots afin de spécialiser ces plongements pour la tâche (*trainable*). Ces deux méthodes de *rétropropagation* sont utilisées dans nos expérimentations.
Nous apprenons un modèle pour chaque corpus, en utilisant en entrée soit les plongements appris sur le corpus *spécialisé*, soit les plongements appris sur Wikipédia (*générique*), et avec les deux modes de *rétropropagation*. Cela représente ainsi 12 modèles distincts, dont nous avons adapté les hyperparamètres à l'aide du corpus DEV.

Métriques utilisées. Nous utilisons le F1-score, qui est la moyenne harmonique du rappel et de la précision. Notre classification étant en classes multiples, nous reportons la moyenne des résultats obtenus individuellement pour chaque classe. Les deux méthodes les plus courantes sont les moyennes *Micro* et les moyennes *Macro*. Les *Micro*-moyennes accordent autant de poids à chacun des documents, tandis que les *Macro*-moyennes ne pondèrent que par le nombre de classes. Ainsi, dans le cas d'un corpus comportant des classes disproportionnées, le modèle obtiendra facilement une *Micro*-moyenne équivalente à la proportion de la classe la plus probable. Nos classes étant disproportionnées, nous utilisons la mesure *Macro*-moyenne.

Significativité des résultats. Notre apprentissage est non déterministe. Ainsi, pour garantir la pertinence des résultats reportés, les expérimentations sont réalisées 10 fois pour chaque configuration. Nous calculons ensuite des intervalles de confiances à 95% pour chaque métrique pour toutes les expériences. Au total, nous avons construit 120 modèles de classification pour l'ensemble de nos corpus, 10 pour chacune des 12 configurations décrites dans le paragraphe précédent.

3 Corpus utilisés

Nous disposons de deux corpus de documents en langue de spécialité annotés en classes : le corpus SNCF (Section 3.1) et le corpus OHSUMED (Section 3.2). Nous utilisons le corpus Wikipédia comme un corpus en langue non spécialisée (Section 3.3).

3.1 Le corpus SNCF

Il s'agit d'un corpus de documents textuels introduit par Dugué *et al.* (2019), en français, privé, appartenant à SNCF. Les documents ont été rédigés par des agents SNCF ou des acteurs du domaine ferroviaire. Ce corpus contient 7 255 documents techniques sur deux domaines : les ressources humaines (définition des différents régimes indemnitaires ou de congés, gratifications, bilan social, etc.) et les opérations (description des manoeuvres dans les gares, consignes de desserte des trains, service de circulation, etc.). Initialement au format PDF, ces documents ont été convertis en documents texte à l'aide de l'outil PDFTOTEXT. Au total, 19 millions d'unités lexicales sont présentes au sein de ce corpus, pour un vocabulaire de 77 mille mots.

Afin de mieux structurer son fonds documentaire, SNCF possède un système de classification hiérarchique de ces documents. Dans nos expérimentations, nous considérons deux niveaux de classification différents. Le premier niveau contient 8 classes, indiquant le thème général du document. Nous notons que la répartition des documents selon ces huit classes est déséquilibrée. En effet, 4 de ces 8 classes représentent 99.8% des documents. Le second niveau de classification, quant à lui, caractérise le sous-thème du document. Celui-ci est composé de 49 classes, avec une distribution très déséquilibrée des documents au sein de ces classes : 97.2% des documents sont représentés par seulement 11 classes.

Nous divisons le corpus SNCF en 3 sous-corpus de façon stratifiée : entraînement TRAIN, développement DEV et TEST, selon les proportions respectives suivantes : 0.60, 0.20, 0.20.

3.2 Le corpus OHSUMED

Le corpus OHSUMED (Hersh *et al.*, 1994) est un corpus public extrait de la base de données MEDLINE, disponible librement et composé de documents textuels en anglais. Ce corpus contient 23 166 abstracts d'articles scientifiques relatifs à différents types de maladies (maladies cardiovasculaires, infections bactériennes, maladies du système digestif, etc.). Ces abstracts ont été rédigés par et à destination de médecins et chercheurs en médecine.

Le corpus contient 23 classes, 16 classes couvrent 90% des documents du corpus. Au total, 4 millions d'unités lexicales sont présentes dans le corpus pour un vocabulaire de 49 mille mots.
Les données du corpus OHSUMED sont séparées en 3 sous-corpus : entraînement TRAIN, développement DEV et TEST selon les proportions décrites dans Joachims (1998) : 0.45, 0.05, 0.50.

3.3 Corpus génériques

Les documents des corpus spécialisés à notre disposition sont en français (SNCF) et anglais (OHSUMED). Ainsi, il est nécessaire de disposer également de deux corpus non *spécialisés* composés pour l'un de documents en français et pour l'autre de documents en anglais. Nous avons pour cela utilisé deux corpus provenant de *dumps* Wikipédia. Un *dump* de Wikipéda est une sauvegarde complète de toute la base de données du site à un instant t. Ces données sont par la suite prétraitées, afin de ne garder que le texte. Ces corpus sont, en comparaison avec les corpus SNCF et OHSUMED, considérés en langue courante : leur contenu se veut non *spécialisé*, et n'est pas restreint à un domaine en particulier.

Le premier *dump* provient de la version anglaise de Wikipédia, enregistrée en mars 2006. Il s'agit du corpus TEXT8 [2]. Celui-ci n'est composé que des 100 premiers méga-octets du *dump* original au format texte. Il contient ainsi 17 millions d'occurrences de mots, et un vocabulaire de 253 mille mots. Le second provient d'un *dump* du Wikipédia français de 2015. Ce corpus s'appelle FR_WIKI_NONLEM (Fauconnier, 2015). Il contient 600 millions d'occurrences de mots, avec un vocabulaire de 191 mille mots.

Afin d'uniformiser les noms de ces corpus, nous nommons par la suite le corpus non spécialisé anglais Wiki-EN et le corpus non spécialisé français Wiki-FR.

4 Expériences et Résultats

4.1 Mesure de la couverture du corpus par dictionnaire

Les dictionnaires choisis pour étudier le vocabulaire d'après l'approche décrite Section 2.1 sont : un dictionnaire français (Pythoud, 1998) pour SNCF, et un dictionnaire anglais issu du modèle de langage EN_CORE_WEB_LG de SPACY (Honnibal & Montani, 2017) pour le corpus OHSUMED. Le dictionnaire français est extrait du corpus Gutenberg et possède un nombre de mots dans son vocabulaire trois fois supérieur au dictionnaire en anglais issu du modèle de langage de SPACY.

	Dictionnaire Français		Dictionnaire Anglais	
	SNCF	Wiki-FR	Ohsumed	Wiki-EN
Couv_{occ}	93.2%	93.8%	96.8%	98.8%
Couv_{voc}	32.6%	72.7%	60.4%	54.8%

TABLE 1 – Couverture du vocabulaire des corpus par deux dictionnaires *génériques*.

Les comparaisons des couvertures des unités lexicales des corpus spécialisés et corpus non spécialisés par rapport aux dictionnaires sont données dans la Table 1.
La même expérience a été réalisée en utilisant le vocabulaire appris lors de la construction des plongements sur les corpus non spécialisés. Ce vocabulaire ne contient que les mots apparaissant au moins 2 fois pour le corpus Wiki-FR, 3 pour Wiki-EN. Ces résultats sont présentés dans la Table 2.

2. Matt Mahoney 2006 ; About the Test Data – http://mattmahoney.net/dc/textdata.html

	Plongements Wiki-FR		Plongements Wiki-EN	
	SNCF	Wiki-FR	Ohsumed	Wiki-EN
Couv_{occ}	91.6%	99.9%*	93.3%	98.9% *
Couv_{voc}	34.0%	88.3%*	47.7%	39.4%*

TABLE 2 – Couverture du vocabulaire des corpus par les plongements de mots *génériques*. Les valeurs notées d'un astérisque* sont directement liées au nombre minimum d'occurrences défini pour qu'un mot soit comptabilisé dans l'espace de représentation.

D'après les calculs de couverture, le corpus OHSUMED est bien couvert (vocabulaire et nombre d'occurrences) par le dictionnaire et les plongements lexicaux *génériques* au regard des résultats obtenus pour le corpus non spécialisé (Wiki-EN). Il semble donc que le corpus OHSUMED comporte principalement du vocabulaire commun. Cette part plus importante de vocabulaire commun peut-être liée au fait que le corpus est composé d'abstracts scientifiques contenant les termes principaux utilisés dans chaque article. Concernant le corpus SNCF, la couverture du vocabulaire est faible au regard des résultats obtenus pour le corpus non spécialisé (Wiki-FR) aussi bien lors de la comparaison au dictionnaire que celle avec les plongements lexicaux. Le vocabulaire SNCF semble donc *spécialisé*. Par ailleurs, on observe que le vocabulaire du corpus non spécialisé français (Wiki-FR) est mieux couvert que celui du corpus non spécialisé anglais (Wiki-EN), alors que la couverture en occurrences est plus forte sur le corpus non spécialisé anglais (Wiki-EN) (Table 1). Cela peut provenir de la différence de taille des deux corpus.

Exemples de mots hors vocabulaires les plus fréquents. Les mots hors vocabulaires du dictionnaire les plus fréquents dans le corpus SNCF sont surtout des acronymes, *spécialisés* du domaine SNCF : *AC*, *PN*, *SGTC*, etc. Ceux non couverts issus du corpus Wiki-FR portent eux sur des noms propres : *France*, *Europe*, *Charles*, *Paul*, etc. En ce qui concerne le corpus OHSUMED, les mots non couverts par le dictionnaire anglais portent sur des noms de maladies ou de termes médicaux : *postoperatively*, *immunohistochemical*, ou encore *histopathologic*. Les unités lexicales non couvertes du corpus Wiki-EN sont en grande majorité des commandes LaTeX. Cela semble montrer un problème de nettoyage des données du corpus anglais *générique* utilisé.
Nous retrouvons globalement les mêmes termes lors de l'analyse en utilisant un dictionnaire extrait des plongements lexicaux. Ces unités lexicales non couvertes sont donc en majorité des termes propres au domaine du corpus *spécialisé* en question.

4.2 Mesure de diversité lexicale

Corpus	MTLD
SNCF	39.7
Wiki-FR	42.9
Wiki-EN	54.2
OHSUMED	66.5

TABLE 3 – Scores de diversité lexicale MTLD pour chaque corpus.

Nous calculons dans un second temps le score de diversité lexicale *MTLD* (Tab. 3) décrit dans la Section 2.1 : plus ce score est élevé, plus le vocabulaire employé dans les documents du corpus est varié. Tout d'abord, la diversité lexicale du corpus SNCF est faible, ce corpus devrait donc être plus aisément couvert par le dictionnaire. Or, nous montrons (Tab. 1) que le vocabulaire du corpus SNCF n'est pas bien couvert par le dictionnaire français (pourtant 3 fois plus grand que celui de l'anglais). Cela semble indiquer que de nombreux termes spécialisés sont employés dans le corpus SNCF. Dans le cas du corpus OHSUMED, la diversité lexicale est importante. Néanmoins, la couverture du vocabulaire du corpus OHSUMED par le dictionnaire anglais est bonne. Le vocabulaire employé est donc peut-être moins *spécialisé* que celui des documents du corpus SNCF. Le corpus SNCF semble donc plus *spécialisé* en termes de lexique

qu'OHSUMED, de par son vocabulaire moins couvert et moins diversifié.

4.3 Mesure de lisibilité des corpus

Les scores de lisibilité ont été calculés à l'aide de la bibliothèque python PY-READABILITY-METRICS [3]. La mesure SMOG définie Section 2.1 pour le corpus SNCF est ensuite convertie pour obtenir le score équivalent en anglais (Éq. 4). Les indices SMOG et ARI sont présentés pour les deux corpus dans la table 4. Le corpus SNCF et le corpus OHSUMED obtiennent des scores SMOG et ARI aussi élevés, ce qui indique qu'ils sont difficilement lisibles. Le score SMOG pour le corpus SNCF est converti en anglais (score brut en français : $\text{SMOG}(\text{FR}) = 22$). La mesure ARI ne possède pas de coefficients de conversion du français vers l'anglais. Nous présentons ainsi les résultats bruts pour ARI. À titre de comparaison, nous donnons les scores SMOG et ARI pour la Déclaration universelle des droits de l'homme (*DUDH*) ainsi que SMOG pour plusieurs ouvrages en anglais. On observe que les phrases sont en moyenne plus longues dans le cas du corpus SNCF (≈29 mots/phrase) mais les mots contiennent en moyenne moins de syllabes (≈1.5 syllabe/mot). Le nombre de syllabes est en moyenne plus important pour le corpus OHSUMED (≈1.9 syllabe/mot) ce qui explique que le score SMOG plus important. Les scores de lisibilité semblent montrer que les corpus SNCF et OHSUMED sont bien spécialisés par le style utilisé dans la rédaction des documents qui les composent (longueur des phrases et nombre de syllabes des termes utilisés).

4.4 Résultats de classification

Nous présentons dans les Figures 2 et 3 les résultats de la classification en *Macro*-F1-score en au cours de l'apprentissage sur le corpus SNCF, tandis que la Figure 4 concerne la classification sur le corpus OHSUMED.
Ces résultats sont présentés sur les corpus de DEV afin de présenter l'évolution de l'apprentissage et non les résultats finaux. En fin d'apprentissage, les résultats obtenus sur les corpus DEV sont proches, et suivent la même tendance que ceux obtenus sur les corpus TEST.

Corpus	SMOG	ARI
SNCF	15.6*	16.4
OHSUMED	17	15.6
DUDH †	13	9.3
Don Quichote‡	11.24	-
La Bible‡	9.35	-
Blanche-Neige‡	6.72	-

TABLE 4 – Scores SMOG et ARI pour les corpus SNCF (FR) et OHSUMED (EN). *Conversion du score FR-EN, † résultats issus de Jakobsen & Skardal (2007), ‡ résultats issus de Contreras *et al.* (1999).

4.4.1 Sur le corpus SNCF

Nous pouvons tout d'abord observer qu'apprendre un classifieur en utilisant les plongements de mots *spécialisés*– ceux appris sur SNCF – sans faire la *rétropropagation* sur ces plongements conduit aux plus mauvais résultats (courbes discontinues avec des marqueurs losanges). Ceux-ci convergent moins vite, et obtiennent des résultats inférieurs aux autres.
Ensuite, les classifieurs entraînés avec les plongements *génériques* (appris sur le corpus Wikipédia français, marqueurs en forme de croix) convergent globalement rapidement, mais pas aussi bien que les modèles de classification appris avec les représentations spécialisées qui bénéficient de l'option *trainable*. Ces modèles convergent plus rapidement vers un meilleur *Macro*-F1-score. Dans le cas du premier niveau de classification, ce modèle n'apporte cependant pas de résultats finaux meilleurs que ceux appris avec les plongements lexicaux non *spécialisés* (marqueurs en forme de croix), ceux-ci

3. https://pypi.org/project/py-readability-metrics/

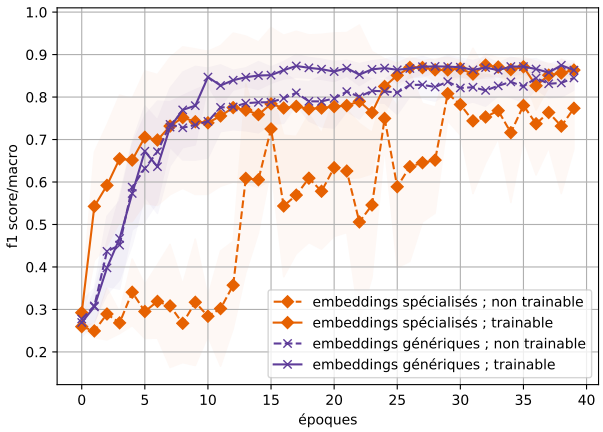


FIGURE 2 – *Macro*-F1-score sur le corpus DEV au cours de l'apprentissage du premier niveau de la classification du corpus SNCF. Intervalle de confiance à 95%.

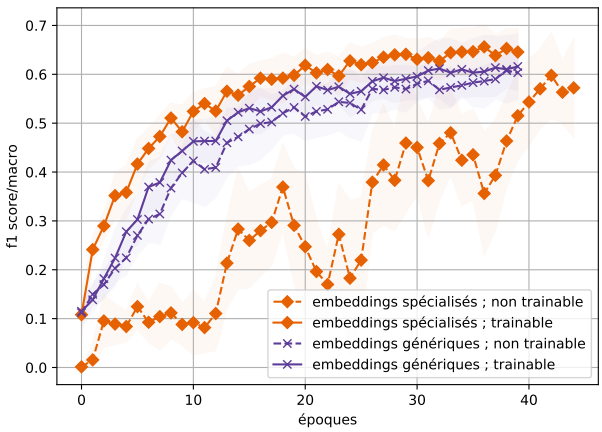


FIGURE 3 – *Macro*-F1-score sur le corpus DEV au cours de l'apprentissage du second niveau de la classification du corpus SNCF. Intervalle de confiance à 95%.

sont équivalents.
Il faut donc faire la rétropropagation de l'erreur en *trainable*. Apprendre des représentations *spécialisées* n'apporte pas de gain notable, il semble donc préférable d'utiliser des plongements pré-appris.

4.4.2 Sur le corpus OHSUMED

Les modèles appris avec les plongements de mots *spécialisés* obtiennent des résultats significativement meilleurs que ceux obtenus avec des plongements *génériques*. Cette différence est bien plus prononcée sur le corpus OHSUMED qu'elle ne l'est sur le corpus SNCF. Pourtant, le corpus OHSUMED est cinq fois plus petit que le corpus SNCF, et apprendre une représentation correcte du vocabulaire de ce corpus semble intuitivement plus difficile que pour le corpus SNCF.
D'après l'indicateur de *couverture* (Section 2.1), le corpus SNCF semble plus *spécialisé* que le corpus OHSUMED, et pourtant c'est pour OHSUMED qu'il est plus pertinent d'apprendre des plongements *spécialisés*. Ces résultats vont donc à l'encontre des intuitions que nous pouvons avoir à l'issue de la Section 2.1. En revanche, ces résultats confirment qu'il est préférable de mettre à jour les plongements de mots lors de l'apprentissage de la tâche de classification (courbes continues *trainable*).

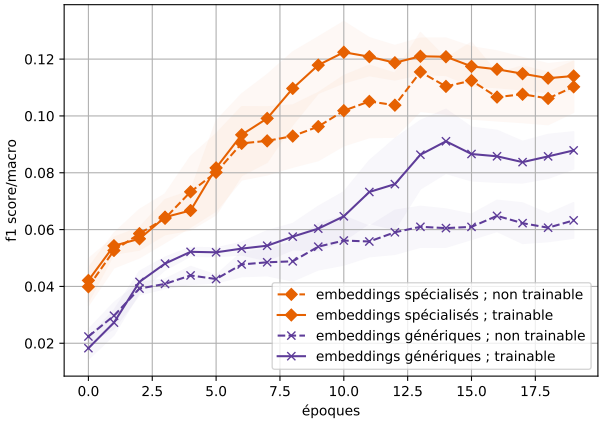


FIGURE 4 – *Macro*-F1-score sur le corpus DEV au cours de l'apprentissage de la classification du corpus OHSUMED. Intervalle de confiance à 95%.

5 Conclusion et perspectives

Dans cet article, nous cherchons à déterminer quels types de plongements utiliser lorsque l'on considère des documents issus d'un corpus en langue de spécialité. En effet, il est possible d'utiliser des plongements préappris sur un corpus plus volumineux tel que Wikipédia, ou bien d'apprendre des plongements sur le corpus *spécialisé* de taille réduite. Pour mener à bien nos expérimentations, nous utilisons deux corpus en langue de spécialité, OHSUMED et SNCF. Nous détaillons dans un

premier temps quelques indicateurs pour tenter d'estimer le degré de spécialité de ces corpus, par ex. la *couverture* du vocabulaire, la diversité du lexique et des scores de lisibilité. On constate que le vocabulaire du corpus SNCF semble particulièrement *spécialisé*, il est très peu couvert par les dictionnaires alors qu'il est moins varié. Le corpus OHSUMED est lui plutôt bien couvert. Par ailleurs, nous montrons avec les scores de lisibilité que ces corpus semblent complexes, c.-à-d. construits avec des phrases longues et de nombreux mots polysyllabiques, en particulier pour OHSUMED. Enfin, le corpus OHSUMED est cinq fois plus petit que le corpus SNCF au regard du nombre d'unités lexicales, il semble donc plus difficile d'obtenir des plongements de bonne qualité sur ce corpus.
Nous apprenons ensuite à catégoriser les documents de ces corpus en utilisant en entrée de nos classifieurs soit des plongements appris sur des corpus Wikipédia (*générique*), soit des plongements appris sur le corpus considéré *spécialisé*. Il s'agit de voir quels types de plongements sont les plus efficaces pour résoudre ce problème. Les résultats confirment les travaux de Wang *et al.* (2018) sur les corpus médicaux : dans le cas d'OHSUMED, les résultats sont meilleurs avec des plongements appris sur ce même corpus. Pourtant, ce corpus est plus petit que SNCF et son vocabulaire mieux couvert avec un dictionnaire classique. En revanche, dans le cas de SNCF, les plongements appris sur ce dernier ne sont pas plus performants que ceux appris sur Wikipedia. Le corpus est pourtant moins bien couvert, et de plus grande taille.

Il est donc difficile de conclure avec ces résultats préliminaires. Nous pouvons tout de même dire que des plongements appris sur un corpus non spécialisé permettent d'obtenir une *baseline* de bonne qualité, même si dans le cas d'OHSUMED, réapprendre les plongements améliore les résultats. Par ailleurs, il est nécessaire de faire la *rétropropagation* jusqu'à la couche de plongements lexicaux. Spécialiser les plongements pour la tâche garantit de bons résultats pour la classification. Enfin, la *couverture*, la *lisibilité* ou la *taille* du corpus ne semblent pas être des indicateurs suffisants pour pouvoir décider s'il faut ou non apprendre des plongements *spécialisés*. Par exemple, de façon contre-intuitive, même si l'on dispose de peu de cooccurrences sur OHSUMED, il vaut mieux apprendre des plongements sur ce corpus pour classifier ses documents. Et même si le vocabulaire de SNCF semble plus *spécialisé*, contre-intuitivement, des plongements appris sur un corpus non spécialisé obtiennent de bons résultats.

En perspective, il s'agirait bien entendu de reproduire cette étude sur d'autres corpus. Dans notre étude, nous disposons de deux corpus en langue différente, et cela peut fragiliser nos conclusions : les dictionnaires sont différents, les corpus Wikipédia également. De plus, pour mieux caractériser la spécificité des corpus, nous pourrions utiliser d'autres indicateurs. Par exemple, la couverture des acronymes et abréviations à l'aide d'un lexique, la distribution des étiquettes morphosyntaxiques (POS), ou encore la distribution des transitions entre ces étiquettes semblent être des estimateurs utiles (Campbell & Johnson, 2001). Par ailleurs, il serait également intéressant d'étudier les effets possibles de la terminologisation en nous basant sur des ontologies spécialisées au domaine des corpus traités. Enfin, il serait également intéressant de comparer les espaces appris par la méthode de plongements afin d'étudier de manière qualitative ce qui différencie les plongements appris sur les corpus *spécialisés*, des plongements appris sur les corpus non *spécialisés*.

6 Remerciements

Ces travaux ont été financés dans le cadre du partenariat entre SNCF I&R et le LIUM. Nous remercions particulièrement L. Lefeuvre de nous avoir autorisé à mener ces travaux et l'équipe du LIUM (N. Dugué, N. Camelin et J. Wottawa) pour leurs conseils avisés.

Références

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*. DOI : 10.1162/tacl_a_00051.

CABRÉ M. T. (2002). Terminologie et dictionnaires. *Meta*. DOI : 10.7202/002182ar.

CAMPBELL D. A. & JOHNSON S. B. (2001). Comparing syntactic complexity in medical and non-medical corpora. In *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, p. 90–94 : American Medical Informatics Association.

CHARNOCK R. (1999). Les langues de spécialité et le langage technique : considérations didactiques. *ASp*. DOI : 10.4000/asp.2566.

CHUJO K. & UTIYAMA M. (2006). Selecting level-specific specialized vocabulary using statistical measures. *System*, **34**(2), 255–269. DOI : 10.1016/j.system.2005.12.003.

CONDAMINES A. (1997). Langue spécialisée ou discours spécialisé ? In L. LAPIERRE, I. OORE & H. RUNTE, Éds., *Mélanges de linguistique offerts à Rostislav Kocourek*, p. 171–184. Les presses d'Alfa. HAL : halshs-01380935.

CONTRERAS A., GARCÍA-ALONSO R., ECHENIQUE M. & DAYE-CONTRERAS F. (1999). The SOL formulas for converting SMOG readability scores between health education materials written in Spanish, English, and French. *Journal of Health Communication*. DOI : 10.1080/108107399127066.

CRESSOT M. & JAMES L. (1996). *Le Style et ses Techniques*. Presses Universitaires de France - PUF.

DUGUÉ N., CAMELIN N., LEFEUVRE L., LI X., REUTENAUER C. & VAUDAPIVIZ C. (2019). Apprentissage et évaluation de plongements lexicaux sur un corpus SNCF en langue spécialisée. In *Extraction et Gestion des Connaissances*, Metz, France. HAL : hal-01982661.

EL BOUKKOURI H., FERRET O., LAVERGNE T. & ZWEIGENBAUM P. (2019). Embedding Strategies for Specialized Domains : Application to Clinical Entity Recognition. p. 295–301. DOI : 10.18653/v1/p19-2041.

FAUCONNIER J.-P. (2015). French Word Embeddings.

GH M. (1969). SMOG grading : A new readability formula. *Journal of Reading*.

HAHNLOSER R., SARPESHKAR R., MAHOWALD M., DOUGLAS R. & SEUNG H. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, **405**, 947–51. DOI : 10.1038/35016072.

HERDAN G. (1960). *Type-token mathematics : A textbook of mathematical linguistics*, volume 4. Mouton.

HERSH W., BUCKLEY C., LEONE T. J. & HICKAM D. (1994). OHSUMED : An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, SIGIR '94, p. 192–201, New York, NY, USA : Springer-Verlag New York, Inc. DOI : 10.1007/978-1-4471-2099-5_20.

HINTON G. E., SRIVASTAVA N., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arxiv.org*.

HONNIBAL M. & MONTANI I. (2017). spaCy 2 : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

HUANG E. H., SOCHER R., MANNING C. D. & NG A. Y. (2012). Improving word representations via global context and multipleword prototypes. In *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*.

JAKOBSEN T. & SKARDAL T. (2007). Readability index. *Agder University*.

JOACHIMS T. (1998). Text categorization with support vector machines : Learning with many relevant features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI : 10.1007/s13928716.

KIM Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1746–1751, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1181.

KINGMA D. P. & BA J. L. (2015). Adam : A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240. DOI : 10.1093/bioinformatics/btz682.

LIU Q., JIANG H., WEI S., LING Z. H. & HU Y. (2015). Learning semanticword embeddings based on ordinal knowledge constraints. In *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*. DOI : 10.3115/v1/p15-1145.

MCCARTHY P. M. (2006). An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD). *Dissertation Abstracts International Section A : Humanities and Social Sciences*.

MCCARTHY P. M. & JARVIS S. (2010). MTLD, vocd-D, and HD-D : A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*. DOI : 10.3758/BRM.42.2.381.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

PALTRIDGE B. & STARFIELD S. (2016). English for specific purposes. In *Handbook of Research in Second Language Teaching and Learning*, volume 3, p. 56–67. Taylor and Francis. DOI : 10.4324/9781315716893.

PYTHOUD C. (1998). Français-GUTenberg : un nouveau dictionnaire français pour ISPELL. Problèmes résolus et intégration de contributions extérieures. *Cahiers GUTenberg*. DOI : 10.5802/cg.237.

SCHMITT D. (2002). Learning Vocabulary in Another Language. I.S.P. Nation. *ELT Journal*. DOI : 10.1093/elt/56.1.91.

SENTER R. J. & SMITH E. A. (1967). *Automated readability index*. Rapport interne, CINCINNATI UNIV OH.

TORRUELLA J. & CAPSADA R. (2013). Lexical Statistics and Tipological Structures : A Measure of Lexical Richness. *Procedia - Social and Behavioral Sciences*. DOI : 10.1016/j.sbspro.2013.10.668.

VAN DER YEUGHT M. (2016). Protocole de description des langues de spécialité. *Recherche et Pratiques Pedagogiques en Langues de Specialite - Cahiers de l'APLIUT*. DOI : 10.4000/apliut.5549.

WANG Y., LIU S., AFZAL N., RASTEGAR-MOJARAD M., WANG L., SHEN F., KINGSBURY P. & LIU H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*. DOI : 10.1016/j.jbi.2018.09.008.

WEIGEND A. S., RUMELHART D. E. & HUBERMAN B. A. (1991). Generalization by Weight-Elimination with Application to Forecasting. In R. P. LIPPMANN, J. E. MOODY & D. S. TOURETZKY, Éds., *Advances in Neural Information Processing Systems 3*, p. 875–882. Morgan-Kaufmann.

YAMADA I., ASAI A., SAKUMA J., SHINDO H., TAKEDA H., TAKEFUJI Y. & MATSUMOTO Y. (2018). Wikipedia2Vec : An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia.

ŘEHŮŘEK R. & SOJKA P. (2010). Software framework for topic modelling with large corpora. p. 45–50. DOI : 10.13140/2.1.2393.1847.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Représentation vectorielle de paires de verbes pour la prédiction de relations lexicales

Etienne Rigaud[1, 2]
(1) LORIA, 615 Rue du Jardin-Botanique, 54506 Vandœuvre-lès-Nancy
(2) École des Mines de Nancy, 54000 Nancy
etienne.rigaud6@etu.univ-lorraine.fr

RÉSUMÉ

Dans cet article, nous proposons un modèle de représentations vectorielles de paire de mots, obtenues à partir d'une adaptation du modèle Skip-gram de Word2vec. Ce modèle est utilisé pour générer des vecteurs de paires de verbes, entraînées sur le corpus de textes anglais Ukwac. Les vecteurs sont évalués sur les données ConceptNet & EACL, sur une tâche de classification de relations lexicales. Nous comparons les résultats obtenus avec les vecteurs paires à des modèles utilisant des vecteurs mots, et testons l'évaluation avec des verbes dans leur forme originale et dans leur forme lemmatisée. Enfin, nous présentons des expériences où ces vecteurs paires sont utilisés sur une tâche d'identification de relation discursive entre deux segments de texte. Nos résultats sur le corpus anglais *Penn Discourse Treebank*, démontrent l'importance de l'information verbale pour la tâche, et la complémentarité de ces vecteurs paires avec les connecteurs discursifs des relations.

ABSTRACT

Verb-pairs embeddings for discourse relation prediction

This paper proposes a model to obtain vector representations of pairs of words, obtained from an adaptation of the Word2vec Skip-gram Model. This model is used to generate embeddings for pairs of verbs, trained on the english corpus Ukwac. The pair-embeddings are then evaluated on a classification task, where the goal is to predict the lexical relation between the input pair of words. The scores obtained on this task with the pair-embeddings are compared with the scores obtained with individual word-embeddings and with pairs of lemmatized verbs. Finally, the pair-embeddings are used on the discourse relation prediction task, on the *Penn Discourse Treebank* dataset, revealing the relevance of verbs for this task, and the complementarity between the verbs and the discourse connective.

MOTS-CLÉS : Vecteur mot, vecteur relation de verbes, analyse de texte, prédiction de relation du discours.

KEYWORDS: Word embedding, relation embedding between verbs, text analysis, discourse relation prediction.

1 Introduction

Dans le domaine du traitement automatique des langues, l'apprentissage de relations entre deux mots donnés est extrêmement important, pour des tâches telles que répondre à des questions, la recherche d'antonymes ou de synonymes, ou bien simplement pour mieux comprendre comment les langages se construisent. Cela motive les recherches présentées dans cet article, qui ont pour

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 179–192.
HAL : hal-02786199.

objectif de construire des représentations de relations entre une paire de mots, et d'évaluer la qualité de ces représentations dans différentes tâches. Pour ce faire, nous présentons un modèle permettant d'entraîner des représentations vectorielles de relations entre des paires de mots.

Les représentations vectorielles de mots sont très utilisées en traitement automatique des langues depuis les années 1990, l'idée étant que les mots sémantiquement proches se retrouvent proches dans l'espace vectoriel des mots. On retrouve souvent dans la littérature le terme *word embedding* ou "plongement lexical" pour qualifier ces représentations vectorielles. Ces représentations sont fondées sur l'hypothèse distributionnelle (Harris, 1954) : "les lexèmes possédant un contexte linguistique similaire ont un sens similaire.". On peut par exemple obtenir des représentations vectorielles de mots en construisant une matrice de co-occurrences, en comptant le nombre d'occurrences de mots dans le contexte d'autres mots (Turney & Pantel, 2010). L'inconveniant de cette méthode étant néanmmoins la taille importante des vecteurs obtenus, et leurs composantes qui sont majoritairement nulles.

Outre ce modèle distributionnel, les modèles utilisant des réseaux de neurones se sont fortement développés, un des modèles populaires étant Word2vec (Mikolov *et al.*, 2013a). Ce modèle consiste à entraîner une matrice de plongements de mots, qui se trouve être une des matrices de poids du réseau, à travers une tâche précise. Le modèle Skip-gram de Word2vec prend un mot en entrée et cherche à prédire la probabilité qu'a un autre mot d'apparaître dans le contexte du mot donné en entrée. Le modèle utilisé dans cet article est une adaptation du modèle Skip-gram à des paires de mots, qui génère des *embeddings* pour des paires de mots.

Dans un premier temps, nous présentons les précédents travaux sur la représentation de relations sémantiques, puis développons une approche particulière de représentation, appliquée aux relations entre des paires de verbes en anglais. Enfin, les expériences menées ainsi que les résultats sont détaillés pour comparer la qualité de ce modèle à d'autres approches. Le code et les données seront rendus disponibles à la communauté.

2 État de l'art

L'apprentissage de représentations vectorielles pour des relations lexicales est très utile pour de nombreuses tâches, comme l'inférence, la recherche d'analogie ou la classification de relations du discours (Braud & Denis, 2016). Des modèles basés sur la composition de vecteurs mots permettent d'apprendre des relations complexes. En effet, ces vecteurs vérifient des relations algébriques qui traduisent des relations sémantiques. Par exemple $\vec{v}_{France} - \vec{v}_{Paris} + \vec{v}_{Madrid} \approx \vec{v}_{Espagne}$ (Mikolov *et al.*, 2013b).

De cette équation, on peut déduire que $\vec{v}_{France} - \vec{v}_{Paris} \approx \vec{v}_{Espagne} - v_{Madrid} \approx \vec{v}_{est\,capitale\,de}$.

Des travaux récents montrent des modèles de représentations de relations entre plusieurs mots, voire de *sentence embeddings* fonctionnant en composant les vecteurs de chaque mot de la phrase (Weston *et al.*, 2013; Hill *et al.*, 2016). Les vecteurs relations peuvent être appris en adaptant les modèles d'apprentissage de vecteur mot Word2vec, comme cela a été fait dans Chingacham & Paperno (2018) : ces auteurs proposent une adaptation de Skip-gram aux paires de noms, et en entraînant un réseau de mapping avec des vecteurs mots pré-entraînés pour régler le problème des données éparses. Similairement, Jameel *et al.* (2018) ont adapté le modèle GloVe (Pennington *et al.*, 2014) pour apprendre des représentations de relations entre une paire de mot, sans l'utilisation de vecteurs mots pré-entraînés.

Les travaux récents utilisent en grande majorité de l'apprentissage non supervisé pour apprendre les représentations de relations, comme c'est le cas avec les modèles adaptant Word2vec (Joshi *et al.*, 2018; Chingacham & Paperno, 2018). Ces modèles sont en général transférés par la suite sur d'autres tâches, comme de l'inférence ou de l'annotation d'image par exemple. Des recherches récentes ont exploré différents modèles possibles pour entraîner des représentations de relations sémantiques, en cherchant à obtenir les meilleures performances possibles sur ces tâches de transfert. Les modèles de réseaux récurrents *LSTM* ou *LSTM Bi-directionnel* obtiennent par exemple de très bons résultats sur une grande variété de tâches. La difficulté réside dans le choix de l'objectif de ces modèles, afin de créer des vecteurs capturant le maximum d'information sur une unité de texte. De très bons résultats ont ainsi été obtenus en entraînant des réseaux de neurones complexes à faire de l'inférence de textes (Conneau *et al.*, 2017), de la classification de relations du discours explicites (Nie *et al.*, 2019) ou bien de la prédiction de marqueurs du discours (Sileo *et al.*, 2019). Néanmoins, ces modèles complexes nécessitent d'immenses quantités de données, ainsi qu'une très importante puissance de calcul. Comme dans ces approches, nous cherchons à construire une représentation en utilisant une architecture neuronale, mais nous nous distinguons par la simplicité du modèle utilisé.

3 Approche

En général, les représentations sont construites pour un mot ou un fragment textuel comme une phrase ou pour un document entier. Notre approche consiste à construire une représentation pour une paire de mots non adjacents apparaissant dans une phrase : nous espérons ainsi construire une représentation de la relation qui lie les mots considérés. Cette représentation doit prendre en compte le contexte d'apparition de la paire de mots. Nous détaillerons dans le reste de cette section le modèle utilisé pour obtenir des représentations vectorielles de relations entre une paire de verbes. Ce modèle est adapté du travail de A. Chingacham (Chingacham & Paperno, 2018) sur les paires de noms.

3.1 Modèle Skip-gram original

Le modèle Skip-gram est un réseau de neurones à une couche cachée, entraîné à prédire le contexte dans une certaine fenêtre d'un mot donné. Considérons l'exemple ci-dessous, extrait du corpus anglais Ukwac (Ferraresi *et al.*, 2013), avec une taille de fenêtre pour le contexte égale à 5.

Exemple 1 *Sensible guidelines will need to be established for holding case details at a regional level.*

Dans ce modèle, la matrice de poids entre la couche d'entrée et la couche cachée évolue jusqu'à contenir des représentations vectorielles de chaque mot du vocabulaire. Le nombre de neurones dans la couche cachée donne la dimension des vecteurs mots finaux. Cet hyper-paramètre est généralement choisi entre 100 et 1000. La dernière couche du réseau utilise la fonction d'activation Softmax, pour calculer la probabilité qu'a un mot c d'apparaître dans le contexte du mot d'entrée w.

$$p(c \mid w) = \frac{exp(v_c'^T v_w)}{\sum_{c_i=1}^{V} exp(v_{c_i}'^T v_w)} \quad (1)$$

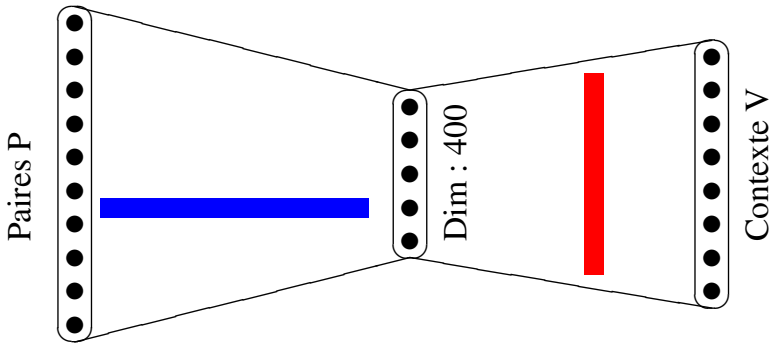


FIGURE 1 – Architecture du réseau utilisé pour générer des vecteurs paires

Où v_w et $v_w^{'}$ sont respectivement les représentations vectorielles de w en tant que mot cible et en tant que contexte.

Enfin, la rétro-propagation s'effectue classiquement en utilisant la fonction de coût de log-vraisemblance négative.

Néanmoins, on peut noter que le coût en calcul de l'équation 1 est très important, à cause de la somme au dénominateur, qui présente un terme pour chaque mot du vocabulaire. Étant donné que la taille du vocabulaire est généralement conséquente, une autre technique proposée par Mikolov *et al.* (2013b), appelée *negative sampling* est nécessaire pour entraîner un modèle avec des coûts de calculs plus raisonnables. Avec le *negative sampling*, le modèle ne doit plus calculer la probabilité $p(c \mid w)$, mais doit prédire si un mot peut apparaître dans le contexte du mot d'entrée. Pour entraîner le modèle, on lui présente donc des exemples "positifs" : des mots qui apparaissent bien dans le même contexte, et des exemples "négatifs" : des mots n'apparaissant pas dans le même contexte. On se retrouve avec un problème de régression logistique pour chaque mot du vocabulaire, ce qui est moins coûteux en terme de calcul.

3.2 Skip-gram pour les paires de mots

Notre but est de générer des représentations vectorielles de relations entre une paire de mots. Nous utilisons une adaptation du modèle Skip-gram classique, en l'appliquant à des paires de mots en entrée du réseau, à la place de mots seuls. Dans ce nouveau modèle, il faut redéfinir l'entrée du réseau, ainsi que la notion de contexte d'une paire de mots. Le modèle que nous utilisons a été pour la première fois implémenté par Chingacham & Paperno (2018) et appliqué aux paires de noms, tandis que nos travaux se concentrent uniquement sur les paires de verbes. En effet nous souhaitions utiliser ce modèle pour diverses tâches de transfert et estimer la pertinence des verbes pour ces tâches. Notre intérêt pour les verbes est dû au fait qu'une partie importante de l'information relationnelle entre deux syntagmes est portée par le verbe. Par exemple, dans la phrase "*Jean **est tombé**, Marie **l'a poussé***.", la relation discursive (causale) entre les deux groupes verbaux est portée par la paire de verbe.

Dans notre modèle, l'entrée du réseau est un encodage *one-hot* d'une paire de mots. Nous avons choisi de ne considérer que le contexte apparaissant entre les deux mots de la paire dans un texte, afin de limiter la taille du vocabulaire du contexte. Néanmoins, de meilleurs résultats pourraient peut-être être obtenus en considérant également les mots apparaissant dans une fenêtre donnée autour de chaque verbe .

Exemple 2 *Sensible guidelines will need to be established for holding case details at a regional level.*

Dans cet exemple, la paire (*will*, *established*) a pour contexte les mots en bleu. Il est important de noter que pour ce modèle, les mots adjacents ne forment pas une paire valide, car ils n'ont pas de contexte, comme la paire (*will*, *need*) dans notre exemple. À part ces redéfinitions, le modèle est exactement le même que le Skip-gram classique avec *negative sampling*.

Avec ce modèle, la fonction de coût est la suivante, où k est le nombre d'exemples négatifs, tirés suivant la distribution P_n :

$$\log \sigma(v_c^{'T} \cdot v_{paire}) + \sum_{i=1}^{k} \mathbb{E}_{c_i \sim P_n} \left(\log \sigma(-v_{c_i}^{'T} \cdot v_{paire}) \right) \tag{2}$$

3.3 Implémentation et construction des vecteurs paires

Le modèle complet décrit dans cet article se découpe en une série d'opérations, composé du traitement d'un important corpus de textes, de l'extraction des paires de verbes, de l'entraînement des représentations vectorielles de paires, puis de l'entraînement d'une matrice de mapping.

3.3.1 Données et pré-traitements

Nous avons utilisé le corpus de texte anglais Ukwac (Ferraresi *et al.*, 2013), contenant 2 milliards de mots. Ce corpus se compose de contenu textuel de pages aspirées du web, et chaque mot est présenté avec sa catégorie grammaticale, ce qui facilite la recherche de verbes.

Avant d'extraire les paires de verbe, nous avons pré-traité le corpus pour enlever certaines informations inutiles pour notre tâche. À l'aide d'un script Python, nous avons transformé le corpus en un document présentant une phrase par ligne où chaque mot est suivi par son étiquette morpho-syntaxique.

Exemple 3 *Portsmouth NP are VBP a DT reminder NN of IN how WRB football NN used VVS to TO be VB.*

Ensuite, pour chaque phrase, les verbes sont identifiés à l'aide de leur étiquette morpho-syntaxique, et les paires de verbes ainsi que les mots de contexte sont comptés. Par ailleurs, afin de réduire la taille du vocabulaire et ne conserver que les paires significatives, seules les paires apparaissant plus de 100 fois sur l'ensemble du corpus sont conservées. De cette manière, le compte final de paires de verbes valides atteint 150 000, et on compte 360 millions de triplets $(verbe_1, verbe_2, contexte)$

Enfin, les exemples d'entraînement positifs et négatifs sont générés pour le *negative sampling*. Nous avons choisi de générer 5 exemples négatifs pour chaque exemple positif.

3.3.2 Entraînement des vecteurs paires

Afin d'entraîner les représentations vectorielles de relations entre deux verbes, nous avons utilisé une adaptation du modèle Skip-gram, détaillée dans la section précédente. Après avoir testé différentes

dimensions de vecteur, comprises entre 50 et 1000, nous avons retenu une dimension de 400, afin d'avoir un bon compromis entre l'information capturée et les temps de calcul. Le modèle a été implémenté à l'aide de la librairie Pytorch. Il implémente une méthode de régularisation ainsi qu'une baisse graduelle du pas d'apprentissage, afin de limiter le sur-apprentissage et d'améliorer la convergence. Cet entraînement a permis de construire un vecteur pour toutes les paires de verbes apparaissant suffisamment souvent dans le corpus. Nous avons entraîné une matrice de mapping afin de construire un vecteur pour toutes les paires de verbes possibles, et ainsi améliorer la couverture des corpus d'évaluation.

3.3.3 Entraînement d'une matrice de mapping

La matrice de mapping est un réseau de neurones prenant en entrée la concaténation de deux vecteurs mots pré-entraînés, et donnant en sortie le vecteur relation entre les deux mots en entrée. L'objectif est de pouvoir obtenir un vecteur relation pour n'importe quelle paire de verbes pour lesquels nous disposons d'un vecteur mot pré-entraîné. Pour les entrées du neurones, nous avons utilisé des vecteurs mots pré-entraînés issus de Baroni *et al.* (2014).

La paire de vecteurs est passée dans un perceptron multi-couches, dont la couche de sortie est la représentation vectorielle de la relation entre les mots d'entrée. Nous avons déterminé empiriquement des valeurs pour les hyper-paramètres et l'architecture du réseau en faisant une évaluation du modèle sur un jeu de données de validation. Finalement nous avons retenu un réseau à une couche cachée, et la fonction d'activation tanh. Nous avons utilisé un pas d'apprentissage de 0.05 et une constante de L2-régularisation de 0.001. Avec cette matrice de mapping, nous pouvons construire un vecteur relation pour toute paire de vecteurs ayant un vecteur mot pré-entraîné.

4 Expériences

4.1 Données

Dans cette section, nous présenterons les corpus utilisés pour l'évaluation des représentations vectorielles de relations entre verbes. Nous présenterons également le corpus utilisé pour la tâche de prédiction de relations discursives.

Pour évaluer la qualité des vecteurs construits avec notre modèle, nous avons utilisé des corpus présentant des paires de mots et la relation correspondante. Nous avions besoin de corpus possédant suffisamment de paires verbe-verbe, donc nous avons utilisé ConceptNet (Speer & Havasi, 2012), un graphe sémantique construit à partir de sources telles que Wikipedia, regroupant des mots dans plus de 300 langues différentes. Dans ce graphe, chaque noeud représente un mot, et chaque arrête représente une relation lexicale. Nous avons extrait de ce corpus uniquement les relations entre deux verbes, ce qui a donné 11 types de relations différents. Nous n'avons pas considéré les paires de verbes ne présentant pas de relations. Nous avons également utilisé le corpus EACL (Nguyen *et al.*, 2017), qui contient des paires de synonymes et d'antonymes, voir Table 1.

Nous avons également utilisé la Penn Discourse Treebank (PDTB) (Prasad *et al.*, 2008), pour la tâche de prédiction de relations du discours. Il s'agit de textes découpés en unités de discours, appelées "arguments", liées par des relations. Notre objectif est d'extraire les verbes de chaque argument et de

Ressource	Type de la relation	Exemples
ConceptNet5	RelatedTo	(eat, feed)
ConceptNet5	Synonym	(jump, leap)
ConceptNet5	MannerOf	(auction, sale)
ConceptNet5	FormOf	(slept, sleep)
ConceptNet5	Antonym	(run, walk)
ConceptNet5	SimilarTo	(compete, win)
ConceptNet5	HasContext	(leave, heading)
ConceptNet5	DistinctFrom	(accept, reject)
ConceptNet5	Entails	(run, move)
ConceptNet5	Causes	(exercise, sweat)
ConceptNet5	DerivedFrom	(paying, pay)
EACL	Synonym	(resurrect, revive)
EACL	Antonym	(love, hate)

TABLE 1 – Jeu de relations et exemples issus de la ressource ConceptNet et du corpus EACL, utilisés pour évaluer nos vecteurs relations

prédire la relation entre les arguments à partir des différentes paires de verbes extraites. Le corpus présente 4 relations de niveau 1, et 16 relations de niveau 2. Ces relations peuvent être implicites en l'absence d'un connecteur discursif, et explicite si les arguments de la relation sont liés par un connecteur.

Le connecteur discursif (*but*, *while*, *because*...) étant porteur de beaucoup d'informations sur le type de relation, nous avons décidé de l'ajouter à notre modèle de vecteur relation. Pour représenter les connecteurs, nous avons assigné à chaque connecteur de la PDTB un index. De cette manière, nous avons pu représenter les connecteurs par des vecteurs *one-hot* de dimension 224, taille du vocabulaire des connecteurs. Par exemple, le connecteur *if* a pour index 85, sa représentation est donc un vecteur de 0, et un 1 à la 85ème position.

Classe (Niveau 1)	Type (Niveau 2)
Comparison	Concession Contrast Pragmatic Concession Pragmatic Contrast
Contingency	Cause Condition Pragmatic Cause Pragmatic Condition
Expansion	Alternative Conjunction Exception Instantiation List Restatement
Temporal	Asynchronous Synchrony

TABLE 2 – Jeu de relations pour les niveaux 1 et 2 dans la PDTB

4.2 Classification de relations

4.2.1 Modèle

Pour évaluer la qualité de notre modèle, nous avons donc utilisé un réseau de neurones réalisant une classification multi-classes prenant en entrée la représentation vectorielle d'une paire de verbes, et essayant de prédire la bonne relation. Les vecteurs en entrée on été construits grâce à la matrice de mapping détaillée en Section 3.3.3.

Nous utilisons un perceptron multi-couches avec une dernière couche Softmax pour réaliser la classification. Classiquement, nous utilisons la fonction de coût de log-vraisemblance à minimiser.

$$-\sum_{y \in train} y \log y_{prediction} \tag{3}$$

Pour le jeu de données PDTB, nous avons tokenisé les phrases avec la librairie Python NLTK, afin de pouvoir trouver et extraire les verbes. Il arrive que des arguments d'une relation contiennent plusieurs verbes. Dans ce cas, nous prenons le vecteur moyen de toutes les paires de verbes entre les arguments. Dans le cas où une relation ne peut pas être représentée par la matrice de mapping, en l'absence des vecteurs pré-entraînés nécessaires, alors nous testons si la paire lemmatisée peut avoir un vecteur par la matrice de mapping. Si ce n'est pas le cas, alors nous donnons à la relation un vecteur moyen, calculé en faisant la moyenne de tout les vecteurs construits par le modèle.

4.2.2 Hyper-paramètres

Pour entraîner le classifieur, nous avons utilisé la descente du gradient stochastique avec une taille de mini-batch de 8, ainsi qu'une régularisation avec norme L2 de constante 0.02, déterminée sur un jeu de données de validation. Par ailleurs, nous avons utilisé la répartition 60% de données à l'entraînement, 10% pour l'ensemble de validation et 30% pour l'ensemble de test. Le classifieur est entraîné sur 100 itérations, avec un pas d'apprentissage décroissant, fixé à 0.01 au départ. Plusieurs architectures de réseaux ont été testées pour la classification, et nous avons finalement gardé un réseau à une couche cachée, de même taille que la couche d'entrée, avec la fonction d'activation ReLU.

Nous avons également testé plusieurs dimensions pour les vecteurs relations, mais peu de différences ont été observées sur le jeu de données de développement. Nous avons finalement retenu une dimension 400, pour pouvoir se comparer au modèle utilisant uniquement la différence de deux vecteurs mots, car les vecteurs mots pré-entraînés dont nous disposions sont de dimension 400.

Dimension du vecteur	50	100	200	300	400	1000
Précision sur l'ensemble de validation	55.08	55.12	56.51	56.70	56.96	57.02

4.2.3 Lemmatisation des verbes

Une intuition au sujet des paires de verbes a été qu'il serait possible d'obtenir de meilleurs résultats en ne construisant que des vecteurs de paires de verbes lemmatisés, c'est à dire dans leur forme neutre. Ce choix a été motivé par le fait que les corpus utilisés présentent en majorité des paires de verbes

sous cette forme. Nous comparerons dans la section résultats les scores obtenus avec des verbes lemmatisés et non lemmatisés.

4.2.4 Exécution

Les unités de calcul nécessaires aux expériences présentées dans cet article, ainsi que l'espace de stockage pour les jeux de données, ont été fournis par Grid5000 (Balouek *et al.*, 2013). L'implémentation est réalisée sur Pytorch, et le code est vectorisé le plus possible pour diminuer les temps d'exécution sur des GPUs.

5 Résultats

Dans cette section seront présentés les résultats de l'évaluation des représentations vectorielles de relations entre deux verbes sur les jeux de données ConceptNet et EACL. Nous donnerons également les résultats obtenus sur la tâche de prédiction des relations du discours sur la *Penn Discourse Treebank*.

5.1 Systèmes de référence

Pour évaluer la qualité de nos résultats, nous les comparerons, pour chaque jeu de données, au deux systèmes de référence suivants :

— La *random baseline*, score obtenu par un modèle prédisant au hasard.
— La *majority baseline*, score obtenu par un modèle prédisant toujours la classe majoritaire du jeu de données.

Les scores présentés sont les scores F1 obtenus sur les données de test. Nous comparerons trois modèles différents de représentation de relations :

— **SkipRel** : Le modèle entraîné par notre variante de Skip-gram pour les paires de verbes. Les vecteurs sont de dimension 400.
— **JustWord** : La différence des deux vecteurs mots de la paire de verbes. Ce modèle utilise des *word embeddings* pré-entraînés de Baroni *et al.* (2014), de dimension 400.
— **RelWord** : La concaténation des deux précédents modèles, résultant en un vecteur de dimension 800.

5.2 Évaluation sur ConceptNet et EACL

Les résultats détaillés de l'évaluation des vecteurs sur la ressource ConceptNet et le corpus EACL sont donnés dans le tableau 3. Nous avons observé que la couverture du jeu de données EACL était mauvaise (moins de 50% des paires de verbes avaient une représentation vectorielle), donc les expériences sur ce jeu de données ont été menées en utilisant la matrice de mapping entraînée pour améliorer la couverture. Par ailleurs, nous avons testé l'évaluation sur les données de ConceptNet avec et sans la matrice de mapping, afin d'observer le gain en performance apporté par les vecteurs construits par la matrice.

Comme expliqué dans la section 4.2.3, nous avons eu l'intuition que les résultats pourraient être améliorés sur la ressource ConceptNet et le corpus EACL en lemmatisant les paires de verbes. Pour cela nous avons lemmatisé les verbes extraits du corpus Ukwac à l'aide de la librairie Python NLTK. Les résultats de l'évaluation avec le modèle entraînés sur des paires de verbes lemmatisés sont également fournis ci-dessous.

Ressource	ConceptNet			EACL	
Modèle	Sans mapping	Mapping	Lemmatisé	Mapping	Lemmatisé
JustWord	50.54	56.46	63.60	**87.32**	**86.85**
SkipRel	48.08	55.70	55.53	70.42	74.36
RelWord	**52.75**	**60.79**	**66.60**	86.88	85.97
RandomBaseline	9.09	9.09	9.09	50	50
MajorityBaseline	34.73	25.3	25.3	50.86	50.86

TABLE 3 – Résultats de l'évaluation des vecteurs paires de verbes

Premièrement, on voit sur ConceptNet que les modèles JustWord et SkipRel obtiennent des scores proches. Néanmoins on observe que le modèle RelWord obtient les meilleurs résultats. Cela signifie que les vecteurs relations que nous avons construits contiennent de l'information différente de l'information contenue par les vecteurs mots individuels. Deuxièmement, les résultats sont améliorés par l'utilisation de la matrice de mapping, car la couverture des données est meilleure. L'utilisation de la lemmatisation sur ConceptNet améliore de façon importante les résultats du modèle JustWord, mais ne semble pas avoir d'impact sur les résultats du modèle SkipRel. Bien que le modèle JustWord obtienne systématiquement de meilleurs scores que notre modèle SkipRel, la concaténation des deux modèles donne les meilleures performances sur ConceptNet, ce qui rend le modèle SkipRel intéressant. Enfin, notre modèle SkipRel donne des résultats très inférieurs à ceux du modèle JustWord sur la tâche de classification d'antonymes et de synonymes, bien que sur cette tâche la lemmatisation des verbes permet une amélioration du score.

En observant l'évolution de la fonction de coût lors de l'entraînement du classifieur sur les données EACL, on se rend compte que le modèle SkipRel fait du sur-apprentissage, ce qui impacte négativement les résultats.

5.3 Prédiction des relations du discours sur la Penn Discourse Treebank

Les scores atteints sur la tâche de prédiction de relations du discours sur les données de la *Penn Discourse Treebank* sont donnés dans le tableau 4.

On distingue les résultats obtenus par les modèles JustWord, SkipRel et RelWord. Dans le cas de relations explicites, c'est à dire les relations qui présentent un connecteur discursif, une représentation *one-hot* de dimension 224 de ce connecteur est concaténée au vecteur SkipRel.

L'évaluation est faite séparément sur les relations implicites et explicites, et nous avons par ailleurs réalisé séparément l'évaluation sur les relations de niveau 2, qui sont plus précises que les relations de niveau 1.

Nous comparons également nos résultats avec les résultats obtenus lors de la *Conll-Shared Task 2016* (Xue *et al.*, 2016).

Type de relation	Implicite		Explicite	
Modèle	Level 1	Level 2	Level 1	Level 2
JustWord	59.71	36.56	62.97	49.81
SkipRel + connecteur (si rel. explicite)	59.01	38.16	74.12	54.01
RelWord	**61.36**	**40.20**	**79.59**	**61.32**
RandomBaseline	25.0	6.67	25.0	6.67
MajorityBaseline	34.73	25.3	35.1	27.3
ConLL Shared Task 2016	-	40.80 [1]	-	78.56

TABLE 4 – Résultats de la tâche de prédiction de relations du discours

On peut sur ce tableau observer l'apport important d'information venant du connecteur, puisque le modèle SkipRel obtient de bien meilleurs résultats sur la prédiction de relations explicites. À nouveau les vecteurs relations que nous avons entraînés semblent être complémentaires avec les *word embeddings* individuels de la paire, car RelWord obtient les meilleurs résultats sur cette tâche, quelque soit le type de relation. Au sujet de la répartition des types de relation sur l'ensemble de test, les relations explicites sont bien équilibrées, les relations de type *Comparison* étant les mieux représentées avec 75% des relations correctement classées dans cette catégorie. Pour les relations implicites, les relations *Temporal* sont sous-représentées, avec moins de 50 exemples, contre plus de 500 relations de type *Expansion*. Malgré tout, 20% des relations *Temporal* sont correctement classées, bien que la relation de type *Comparison* est à nouveau la mieux représentée avec 70% des relations correctement prédites. Nous sommes cependant encore loin des performances état de l'art sur la tâche (voir dernière ligne de 4), ce qui s'explique par le fait que nous ne prenons en compte qu'une partie de l'information disponible dans les arguments. Les scores sont cependant significativement au-dessus de la chance, démontrant que les représentation encodent des informations pertinentes.

Afin de visualiser la qualité de la concaténation des vecteurs relations et du vecteur de connecteur discursif, nous avons représenté ces vecteurs, de dimension 624, en 3D à l'aide d'une analyse en composantes principales (ACP), voir Figure 2. Les couleurs sont les labels des relations explicites de niveau 1. On observe que des groupes se forment, alors qu'une ACP uniquement sur les vecteurs connecteurs ou sur les vecteurs relations ne permet pas de distinguer de clusters. Les relations de type *Comparison* et *Expansion* se distinguent clairement, et les relations de type *Temporal* et *Contingency* se mélangent. Cela fait sens sémantiquement, car le type *Contingency* correspond aux relations causales, or on retrouve souvent une dimension temporelle dans les relations causales.

D'autre part, nous avons réalisé une réduction de dimension avec l'algorithme t-SNE sur les mêmes vecteurs, cette fois pour les relations de niveau 2 (fig. 3). Les couleurs représentent les labels des relations. À nouveau, des clusters se dessinent entre les types de relations les plus représentés, ce qui tend à démontrer que les représentations relationnelles construites sur les paires de verbes encodent une information pertinente pour la dimension rhétorique.

1. Score obtenu sur un jeu plus large de types de relations

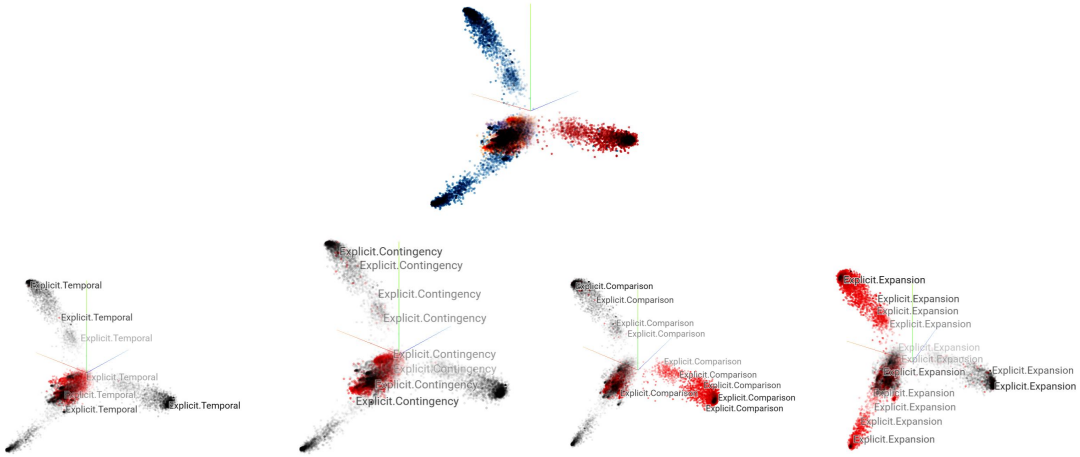


FIGURE 2 – ACP sur les vecteurs SkipRel + Connecteur pour les relations explicites de niveau 1

FIGURE 3 – Résultat de l'algorithme t-SNE sur les vecteurs SkipRel + Connecteur pour les relations explicites de niveau 2

6 Conclusion

Dans cet article, nous avons proposé une approche visant à construire une représentation distribuée, non pour des mots, mais pour des paires de mots dans le but d'encoder leur relation. Nous avons évalué ces représentations sur 2 tâches venant de 3 corpus : la prédiction de relations lexicales (antonymies, synonymies) et la classification de relations discursives (temporelles, causales...). Le modèle utilisé pour la construction de ces vecteurs relations est une variante de Skip-gram, et nos recherches se concentrent sur les paires de verbes. Les résultats montrent que ces vecteurs relations capturent une information différente des vecteurs mots classiques, ce qui rend la composition des deux représentations plus performante sur diverses tâches de transfert. L'évaluation sur la tâche de prédiction des relations du discours a révélé des résultats intéressants sur la façon dont les vecteurs de verbes et les connecteurs logiques se complètent. La visualisation à l'aide d'une analyse en composantes principales en 3 dimensions appuie cette observation. Néanmoins, notre modèle ne permet pas d'approcher les résultats obtenus par des modèles plus complexes, car l'information contenue dans les verbes d'une phrase n'est souvent pas suffisante pour en déduire l'articulation. Des modèles plus avancés capables de générer des représentations vectorielles de phrases entières pourraient compléter notre modèle en permettant la prise en compte de tous les mots de la phrase. La concaténation de ces deux modèles pourrait atteindre de bonnes performances sur la tâche de prédiction de relations du discours.

Remerciements

Je remercie Denis Paperno et Chloé Braud qui ont encadré ce travail, pour leur disponibilité et les précieux conseils qu'ils m'ont fournis.

Nous remercions les relecteurs pour leurs commentaires pertinents.

Références

BALOUEK D., CARPEN AMARIE A., CHARRIER G., DESPREZ F., JEANNOT E., JEANVOINE E., LÈBRE A., MARGERY D., NICLAUSSE N., NUSSBAUM L., RICHARD O., PÉREZ C., QUESNEL F., ROHR C. & SARZYNIEC L. (2013). Adding virtualization capabilities to the Grid'5000 testbed. In I. I. IVANOV, M. VAN SINDEREN, F. LEYMANN & T. SHAN, Éds., *Cloud Computing and Services Science*, volume 367 de *Communications in Computer and Information Science*, p. 3–20. Springer International Publishing. DOI : 10.1007/978-3-319-04519-1_1.

BARONI M., DINU G. & KRUSZEWSKI G. (2014). Don't count, predict ! a systematic comparison of context-counting vs. context-predicting semantic vectors. volume 1, p. 238–247. DOI : 10.3115/v1/P14-1023.

BARONI M. & ZAMPARELLI R. (2010). Nouns are vectors, adjectives are matrices : Representing adjective-noun constructions in semantic space. p. 1183–1193.

BENGIO Y., DUCHARME R., VINCENT P. & JAUVIN C. (2006). *Neural Probabilistic Language Models*, volume 3, p. 137–186. DOI : 10.1007/3-540-33486-6_6.

BRAUD C. & DENIS P. (2016). Learning connective-based word representations for implicit discourse relation identification.

BULLINARIA J. & LEVY J. (2012). Extracting semantic representations from word co-occurrence statistics : Stop-lists, stemming, and svd. *Behavior research methods*, **44**, 890–907. DOI : 10.3758/s13428-011-0183-8.

CHINGACHAM A. & PAPERNO D. (2018). Generalizing representations of lexical semantic relations.

CONNEAU A., KIELA D., SCHWENK H., BARRAULT L. & BORDES A. (2017). Supervised learning of universal sentence representations from natural language inference data.

FERRARESI A., ZANCHETTA E., BARONI M. & BERNARDINI S. (2013). Introducing and evaluating ukwac, a very large web-derived corpus of english.

HARRIS Z. S. (1954). Distributional structure. *Word*, **10(23)**, 146–162.

HILL F., CHO K. & KORHONEN A. (2016). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1367–1377, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1162.

JAMEEL S., BOURAOUI Z. & SCHOCKAERT S. (2018). Unsupervised learning of distributional relation vectors.

JOSHI M., CHOI E. & LEVY O. (2018). pair2vec : Compositional word-pair embeddings for cross-sentence inference.

LENCI A. (2018). Distributional models of word meaning. *Annual Review of Linguistics*, **4**. DOI : 10.1146/annurev-linguistics-030514-125254.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, **26**.

NGUYEN K. A., SCHULTE IM WALDE S. & VU T. (2017). Distinguishing antonyms and synonyms in a pattern-based neural network. DOI : 10.18653/v1/E17-1008.

NIE A., BENNETT E. & GOODMAN N. (2019). Dissent : Learning sentence representations from explicit discourse relations.

PAPERNO D., PHAM N. & BARONI M. (2014). A practical and linguistically-motivated approach to compositional distributional semantics. volume 1, p. 90–99. DOI : 10.3115/v1/P14-1009.

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

PRASAD R., DINESH N., LEE A., MILTSAKAKI E., ROBALDO L., JOSHI A. & WEBBER B. (2008). The penn discourse treebank 2.0.

RITTER S., LONG C., PAPERNO D., BARONI M., BOTVINICK M. & GOLDBERG A. (2015). Leveraging preposition ambiguity to assess compositional distributional models of semantics. p. 199–204. DOI : 10.18653/v1/S15-1023.

SILEO D., VAN DE CRUYS T., PRADEL C. & MULLER P. (2019). Mining discourse markersfor unsupervised sentence representation learning.

SPEER R. & HAVASI C. (2012). Representing general relational knowledge in conceptnet 5. *Proc. of LREC*, p. 3679–3686.

TURNEY P. D. & PANTEL P. (2010). From frequency to meaning : Vector space models of semantics. *Journal of Artificial Intelligence Research*, p. 141–188.

WESTON J., BORDES A., YAKHNENKO O. & USUNIER N. (2013). Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 1366–1371, Seattle, Washington, USA : Association for Computational Linguistics.

XUE N., NG H., PRADHAN S., RUTHERFORD A., WEBBER B., WANG C. & WANG H. (2016). Conll 2016 shared task on multilingual shallow discourse parsing. p. 1–19. DOI : 10.18653/v1/K16-2001.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

TTS voice corpus reduction for audio-book generation

Meysam Shamsi[1]
(1) IRISA, CNRS, 22300 Lannion, France
meysam.shamsi@irisa.fr

ABSTRACT

Nowadays, with emerging new voice corpora, voice corpus reduction in expressive TTS becomes more important. In this study a spitting greedy approach is investigated to remove utterances. In the first step by comparing five objective measures, the TTS global cost has been found as the best available metric for approximation of perceptual quality. The greedy algorithm employs this measure to evaluate the candidates in each step and the synthetic quality resulted by its solution. It turned out that reducing voice corpus size until a certain length (1 hour in our experiment) could not degrade the synthetic quality. By modifying the original greedy algorithm, its computation time is reduced to a reasonable duration. Two perceptual tests have been run to compare this greedy method and the random strategy for voice corpus reduction. They revealed that there is no superiority of using the proposed greedy approach for corpus reduction.

RÉSUMÉ

Réduction du corpus vocal pour la génération de livres audio par TTS

Aujourd'hui, avec l'émergence de nouveaux corpus vocaux, la réduction de voix pour la synthèse de parole TTS expressive devient plus importante. Dans cette étude, une approche de type glouton cracheur pour supprimer des phrases est étudiée. Dans la première étape, en comparant cinq mesures objectives, le coût global du TTS s'est révélé être la meilleure mesure disponible pour l'approximation de la qualité perceptuelle. L'algorithme glouton utilise cette mesure pour évaluer les candidats à chaque étape et la qualité synthétique résultant de la solution en construction. Il s'est avéré que réduire la taille du corpus vocal jusqu'à une certaine durée (1 heure dans notre expérience) ne dégradait pas la qualité synthétique. En modifiant l'algorithme glouton d'origine, il produit une solution en temps raisonnable. Deux tests de perception ont été effectués pour comparer cette méthode gloutonne et la stratégie aléatoire de réduction du corpus vocal. Ils ont révélé qu'il n'y a pas de supériorité dans l'utilisation du glouton proposé pour la réduction du corpus.

KEYWORDS: Text-to-speech, voice corpus, greedy algorithm, perceptual test.

MOTS-CLÉS : Synthèse vocale, corpus vocal, algorithme glouton, test de perception.

1 Introduction

In the Text-to-Speech (TTS) framework, the main reasons to design a small speech database via a corpus reduction approach are generally the database scalability, the recording or the labeling cost reduction, and the elimination of destructive data (Baljekar & Black, 2016).

Several studies could be found in the literature dealing with voice corpus optimization. The main

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 193–204.
HAL : hal-02786200.

purposes of these studies were the pruning of voice corpus in order to respect a parsimony constraint or to extract a more neutral voice, assuming the elimination of expressive voice parts would improve the naturalness of synthetic signals. The idea used by (Krul *et al.*, 2007) was to remove the least selected acoustic units to synthesize signals, using a unit-selection TTS system, for a domain specific application in order to decrease the voice corpus size. This approach was compared with a greedy strategy based on Kullback-Leibler divergence (KLD). The evaluations indicated better achievements for the adaptive domain pruning method than for the KLD based one.

Some studies have been done in order to use voice data stemming from audio-books to provide input for TTS systems. The main goal in this case is to improve naturalness of a neutral voice. An outlier-removal approach has been introduced by (Braunschweiler & Buchholz, 2011; Cooper *et al.*, 2016). The outlier, supposed to involve less natural-sounding speech, has been found out as hypo-articulated utterances and low mean f0 utterances in (Cooper *et al.*, 2016). In (Braunschweiler & Buchholz, 2011), it has been noticed that variety in a speech corpus degrades TTS quality in general task. They discarded sentences based on acoustic features (extreme f0 patterns, too loud or barely audible sentences) and linguistic features (non-neutral style sentences such as quotation, interjections, utterances which start with lowercase, etc). Although TTS systems for a general task need neutral voice corpus (Braunschweiler & Buchholz, 2011; Cooper *et al.*, 2016), the audio-book generation needs an expressive speech synthesis. Found data like available audio-books in the public domain can contain some destructive parts if directly used to build a voice for TTS systems. The strategy consisting on selecting a cleaner subset of speech data may result higher synthetic quality and helps TTS unit selection engine to speed up. In (Baljekar & Black, 2016), two types of errors, misalignment and annotation errors, which degrade TTS quality have been identified to be removed.

However the objectives of previous works were mainly the improvement of the naturalness of synthetic signals, expressiveness plays also a crucial role in audio-book generation. The main aim of this study is audio-book generation using a recorded voice of book reading. Although synthetic speech has less quality than natural one, the audio-book generation could be less costly using a TTS system. Thus, the voice corpus design problem considered here is defined as the script selection for recording in order to use the resulting signals as voice to synthesize the rest of the book. This problem has been investigated in previous studies (Shamsi *et al.*, 2019, 2020) by taking into account the linguistic information. This paper introduces a posterior strategy as a voice corpus reduction : the study is conducted directly from a fully recorded audiobook, instead of its textual content. The voice is a subset of this audio-book and its achievements is assessed by the quality of the synthetic vocalisation of a complementary part of the book. This approach permits to save recording phases and to test several script selections. Moreover, voice corpus reduction methods can profit from acoustic and linguistic information. Analyses of selected voice sub-corpus achievement and content could be helpful for script design based on only linguistic information before recording process in future works.

The original recorded voice is composed of high quality expressive signals which is spoken by a professional speaker. As to offer an adequate expressiveness (from the perspective of later recording phase), the corpus reduction is done at utterance level. Since the final product will be an audio-book mixing natural speech utterances (which compose the voice) and synthesized speech signals, and a bigger voice size generally provides a better synthesized speech quality, the goal of this study is to find the best trade-off between the signal quality of the audio-book and the voice size.

This paper is organized as follows. First, section 2 describes the proposed greedy algorithm and its heuristics to achieve voice corpus reduction. This algorithm needs to evaluate synthetic signals and utterance candidates to remove without requiring human listeners and section 3 investigates an

objective measure for ranking candidates. At last, the evaluation results of the proposed algorithm, in comparison with a random method as baseline, are detailed in section 4.

2 Framework

In order to extract a reduced voice from the original voice corpus well-adapted to synthetically vocalize the content of the unselected sentences, two main requirements are needed : a practical heuristic for selecting a subset of the full corpus and an automatic evaluation method to assess the quality of synthetic signals by using the voice subcorpus. By considering the previous works (François & Boeffard, 2002; Espinosa *et al.*, 2010; Barbot *et al.*, 2015), it has been shown that the greedy algorithm selects portions of a corpus in reasonable time, close to the optimal ones.

The greedy algorithm (with spitting or agglomerative policy) is an iterative strategy and needs a score function to rank candidates. In each step of spitting (resp. agglomerative) greedy process, candidates with minimum (resp. maximum) *utility* are selected to be excluded from (resp. added to) the voice corpus under reduction (resp. construction). The *utility* score of each utterance represents the increasing gain of the richness in the voice corpus when this utterance is kept. This metric will be presented in section 3.

The objective is to extract *voice corpus* from a fully recorded audio-book to be used by TTS. This voice corpus selection should provide the highest synthetic quality for the rest of the book which is called *synthetic part*. The process, described in algorithm 1, is based on a spitting greedy approach and the initial voice corpus to reduce is the whole audio-book. A similar process has been implemented in (Espinosa *et al.*, 2010) : the authors proposed to agglomeratively select utterances which causes the least synthetic quality degradation for a target set of utterances. In the case of our problem, the target set is the rest of the book and is modified by the reduction process.

Algorithm 1: Spitting greedy for audio-book generation

```
1  voice corpus = candidate set = all utterances ;
2  synthetic part = ∅ ;
3  while the candidate set has at least one utterance do
4      for All U_i utterance in the candidate set do
5          Remove U_i from the voice corpus ;
6          Synthesis the synthetic part by using the voice corpus ;
7          if synthesizing of the synthetic part failed then
8              Lock U_i and remove from the candidate set ;
9          else
10             Compute the utility of the U_i based on the quality degradation of synthetic part ;
11         end
12         Add U_i to the voice corpus;
13     end
14     Find U_x with the minimum utility from the candidate set;
15     Remove U_x from the voice corpus and the candidate set and add to synthetic part;
16 end
```

Some utterances in the audio-book contain unique units and a concatenative TTS system cannot find

these units in other utterances. These utterances are locked and should not be removed from voice corpus. At each step, the remaining unlocked utterances in voice corpus compose the *candidate set* for the next step. By removing utterances from voice corpus, a voice subset with a reduced size will be achieved. The spitting greedy process is continued until the *candidate set* is empty.

For each reduction rate, the rest of the book should be synthesized and evaluated in terms of quality degradation. Indeed a small change inside voice corpus could change the synthetic quality of the *synthetic part*. But since synthesizing the whole *synthetic part* each time is computationally expensive, the synthetic signal of selected candidates in each step will be used as approximation of the overall quality degradation. The idea behind this proposition is that the small change of voice corpus by removing an utterance could be ignored and only the quality degradation of final audio-book because of replacing recorded voice of the utterance by its synthetic signal would be taken into account.

In the ideal scenario, in order to investigate the impact of TTS voice corpus reduction on synthetic quality, all combinations of sub-sets should be evaluated perceptually for a given reduction rate. But this is not possible within a reasonable time. In (Espinosa *et al.*, 2010), the measure for ranking utterances and the evaluation of quality are the same (TTS costs). In this way, the quality evaluation process does not need additional computation (algorithm 1 by using the same score in line 10 and 14 follows this idea). This measure will be investigated in section 3. The selection algorithm and the computational problem of the proposed algorithm will be presented in section 4.

The optimization process starts with a full audio-book as a initial corpus. In this study, the initial voice corpus contains 3339 utterances of a French expressive audio-book spoken by a male speaker. The overall length of the speech corpus is 10h44. More information on the annotation process can be found in (Boeffard *et al.*, 2012). The IRISA TTS system (Alain *et al.*, 2017), which is unit selection based, uses voice subcorpus for synthesising.

3 Objective measure for selection

It is impossible to have all synthetic signals evaluated perceptually by listeners. Thus an automatic measure is necessary for quality evaluation of synthetic utterances. This objective measure should be a good approximation of perceptual evaluation. The correlation coefficient or the ranking correlation coefficient could indicate the reliability of an objective measure.

The unit selection TTS costs is used in previous works (Chu & Peng, 2001; Toda *et al.*, 2006; Krul *et al.*, 2007; Espinosa *et al.*, 2010) as the synthetic quality indicator. In this study, the usage of TTS global cost, which is a linear combination of concatenation and target cost, in unit selection TTS is evaluated as the objective measure of synthetic quality. Moreover this objective measure does not need any supplementary computation in the proposed greedy (the result of the line 6 in algorithm 1 can be used directly for line 10).

Beside TTS costs, we propose other objective measures for quality evaluation of synthetic signals. Some measures such as PESQ (Rix *et al.*, 2001) and Dynamic Time Warping (DTW) between two signals need the reference signal. Basically they evaluate similarity between a test signal and its reference. Three DTW based measures are proposed; a DTW between Mel-Frequency Cepstral Coefficients (MFCC) features of the test signal and its natural pair, a DTW between Mel-Generalized Cepstral Coefficients (MGC) features of the test signal and its natural pair, and a DTW between MGC features of test signal which is synthesized signal using voice subcorpus and a reference signal which

is synthesized signal using the whole of voice corpus. The third DTW calculates the degradation quality of an synthetic test signal from the highest possible synthetic quality using the TTS.

3.1 Experimental setup

To investigate the correlation of these objective quality measures with perceptual quality, a listening test (DMOS) is designed. Six different sub-voice corpora of different sizes (75%, 50%, 25%, 10%, 5%, and 1% out of the initial audio-book) are selected randomly. In this experiment, the rest of the book will be synthesized and used for perceptual evaluation. The listeners are asked to evaluate 60 synthetic samples from each synthetic part corresponding to corpus size. By providing the natural voice of each synthetic signal, the quality degradation of synthetic signal are asked on a scale from 1 to 5 (5 means without quality degradation and 1 means the highest degradation).

3.2 Result

The perceptual test resulted in 850 evaluation scores by 17 listeners. A perceptual score is assigned to each sample by getting the average of its scores. Two ranking correlation coefficients (Spearman (Spearman, 1904) and Kendall tau (Kendall, 1948)) and Pearson correlation coefficient (Freedman *et al.*, 2007) are computed between average perceptual score and objective scores of samples. The correlation coefficients between listener scores and 5 objective measures for all voice corpus sizes are compared in table 1.

Objective measures	PESQ	DTW-MGC (Natural ref)	DTW-MFCC (Natural ref)	DTW-MGC (Synthetic ref)	TTS global cost
Pearson C.C.	0.07(p>0.2)	-0.41(p<0.001)	-0.38(p<0.001)	-0.40(p<0.001)	**-0.66(p<0.001)**
Spearman R.C.C.	0.08(p>0.1)	-0.39(p<0.001)	-0.39(p<0.001)	-0.40(p<0.001)	**-0.65(p<0.001)**
Kendall tau R.C.C.	0.05(p>0.1)	-0.28(p<0.001)	-0.28(p<0.001)	-0.28(p<0.001)	**-0.48(p<0.001)**

TABLE 1: Correlation coefficients between objective measures and perceptual evaluation and their p-value.

According to table 1, the TTS global cost has a stronger correlation with perceptual scores than PESQ or DTW on different acoustic features (MFCC, MGC).

While the reported correlation coefficients are calculated on synthetic signals with 6 voice corpus sizes, the mean of perceptual and objective scores on each voice corpus size could reveal more information. The impact of voice corpus length on synthetic quality (with perceptual and objective measures) is investigated. Figure 1 compares the perceptual and objective scores for synthetic utterances with different sub-corpus sizes. The horizontal axis indicates the size of voice subcorpus out of initial voice corpus which is selected randomly.

The increasing trend of MOS score and decreasing trend of TTS global cost for larger voice subcorpus confirm that the quality of synthetic signals will be improved with larger voice corpus. But the perceptual quality of synthetic signals with 25%, 50%, and 75% of the initial corpus (more than 1 hour) are not significantly different. It means that using more data for TTS voice corpus after a threshold would not improve significantly the speech quality.

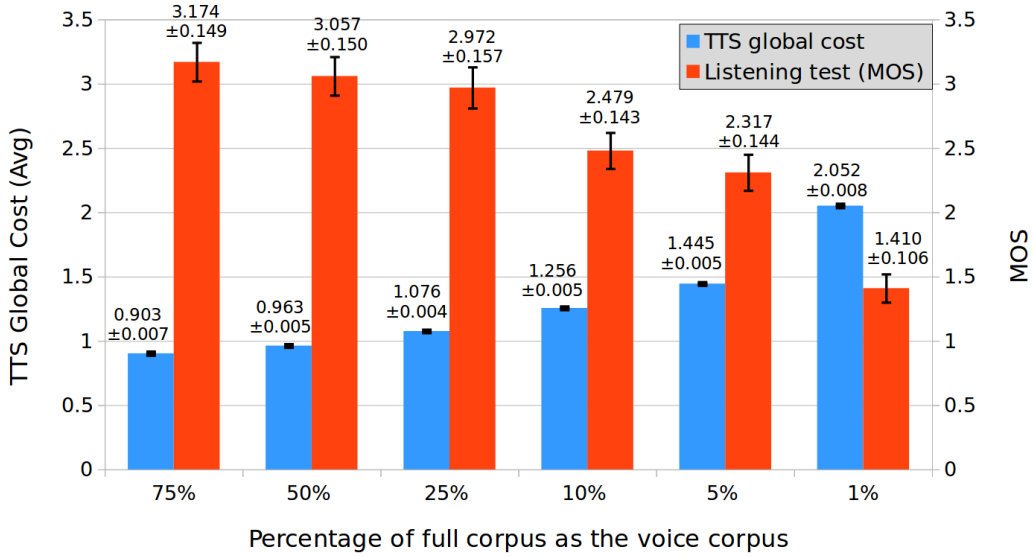


FIGURE 1: The TTS global cost and perceptual score for different voice corpus sizes.

In the remainder of this study, the TTS global cost will be used as an approximation of perceptual quality with the spitting greedy approach, which is evaluated in the next section.

4 Optimization strategy

In this section, a modified greedy process based on algorithm 1 will be compared with a random selection for voice corpus reduction in terms of synthetic quality.

4.1 Modified Greedy algorithm

The computational time of the algorithm 1 makes its use unfeasible for large voice corpora. Calling TTS for synthesizing *the synthetic part* and evaluating its quality are expensive. Consequently, we propose to do two modifications on the original algorithm in order to find a solution in a reasonable time. The first modification is removing bunch of utterances in each step instead of one utterance. And the second one is evaluating the utility of *the candidate set* (algorithm 1, line 10) based on the quality of the synthetic signal of a single utterance instead of the quality degradation of *the synthetic part*. In the following these proposition will be justified.

By analyzing the ranking list of utterances in consecutive steps of the algorithm, it has been observed that the ranking list of candidate utterances does not change a lot from one iteration to the next. Although a simple experiment has showed that this assumption is not completely true, considering the utility of utterances (for being in voice corpus) as a stable rank list helps to get rid of all computational problems. The normalized root-mean-square error (NRMSE) of the TTS global cost for corpus reduction from initial corpus to 70% of initial corpus was 0.012. It shows that, in a big corpus, following the initial ranking list makes the resulting sub-corpus slightly worse than following the original spitting greedy. It leads to a compromise between updating ranking list after each change, which is computationally expensive, and using initial ranking list, which gives a less efficient solution.

We propose to remove a bunch of utterances (100 utterances) based on the ranking list at each step of the spitting greedy. The ranking list is then updated after removing the bunch of utterances. This modification reduces the computational time.

In the second proposition, the computational time of the candidates list evaluation will be reduced. Indeed a change inside the corpus could have an effect on the whole of the *synthetic part*. It means that the *synthetic part*, which U_i has been added to, should be evaluated for U_i's utility in the candidate list. When the evaluation of synthetic part for each U_i candidate is computationally expensive, we propose to consider the TTS global cost of synthesized U_i as an inverse metric for utility. It helps to save synthesizing time of *synthetic part* for evaluating candidates. This idea changes the algorithm 1 by modifying the *synthetic part* to U_i (line 10).

By applying these two modifications on the original greedy algorithm, the computational time will be reduced drastically. The number of utterances in each removal step has a linear relation with computational time. Based on our physical facilities, by removing a bunch of 100 utterances, the experiment will be finished in 2 days, which seems reasonable, and it gives a reduction steps of 3% (or 20 minutes) for voice corpus reduction. Although choosing smaller number of utterances in removal bunch improve the result, it costs computational time.

Comparing the original spitting greedy and modified spitting greedy on a small corpus (334 utterances) shows that the sub-corpus resulting from the proposed greedy synthesizes with higher TTS global cost in a shorter time. It is expected since the modified algorithm is not as efficient as the original one even if it helps to find a solution in a reasonable time.

4.2 Experimental setup

In order to compare perceptually the corpus reduction methods a *test section* (10% of the initial voice corpus) is extracted from the book. Although the main problem in our case is to synthesize the rest of the book, a synthetic part as *test section* would help to compare different methodologies for corpus design. It is assumed that the voice corpus, which is supposed to synthesise the rest of the book, has almost same performance on the *test section*. We assume that since the test part comes from the same book, the synthetic quality of this part can be generalized to the rest of the script. The *test section* is randomly selected as a continuous part with 334 utterances. The remaining part of the audio-book is named the *full corpus*. The *test section* has been synthesized by TTS using 100%, 70%, 40%, 15%, 7%, 3% of the *full corpus*. The voice corpus reduction is done based on the spitting greedy and a random strategy. The initial corpus is the same audio-book as what has been described in previous section (see the end of section 2).

In the following, the evaluation of proposed corpus reduction methods will be detailed. Two perceptual tests are designed to evaluate the quality of signals which are synthesized using resulting voice subcorpora. Based on the previous perceptual test results, the objective measure for ranking utterances in the spitting greedy is the TTS global cost. The first perceptual test is designed to investigate the impact of voice corpus reduction by modified greedy on synthesizing quality. The purpose of the second perceptual test is to compare the performance of proposed greedy and random strategy.

4.3 Synthetic quality degradation by greedy voice reduction

The greedy and random methods provide voice subcorpora with different length to synthesize *test section*. Since the IRISA TTS is unit selection-based, some of the utterances would failed to be synthesized specially in small voice corpus size. After removing these uncommon samples, 70 utterances has been selected randomly. In order to have samples with an acceptable duration, some utterances have been concatenated or cut. More precisely, if the length of selected synthetic signal is less than 4 seconds, the next utterance in script order is concatenated to it. The first 6 seconds of synthetic signals have been cut and used as listening samples. Samples from 6 voice corpus sizes and two corpus reduction methods are used to design a MUSHRA test (Recommendation, 2003). For each step of the perceptual test, the overall quality of 11 synthetic signals, which have been synthesized with different sub-voice corpora, have been evaluated on a scale form 1 to 10 (with one by one increment). Synthetic signals and corresponding natural voice, which have the same script, are available to listeners. The listeners are asked to do 10 steps of MUSHRA test after an introduction step. The estimated time for doing this test is 25 minutes.

This perceptual test has been done by 14 listeners which provides 1441 evaluation scores for synthetic signals. To investigate the impact of corpus size on synthetic signals, the average score for each size/method has been calculated. The figure 2 (left) shows that the average scores for all voice corpus are in a almost same level. It indicates, not only the quality of synthetic signals based on random and greedy strategy dose not have significant different, but also reducing the voice corpus size can not impact on resulted quality at least until 15% of corpus reduction rate. The listeners evaluated 20% of the signals with exact same values.

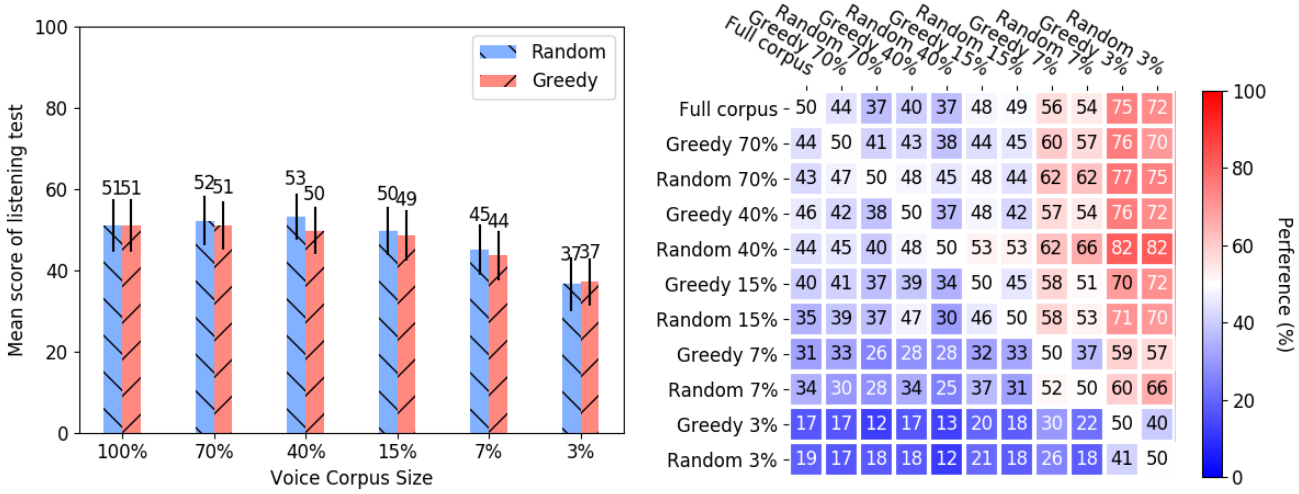


FIGURE 2: The average MUSHRA scores for different voice corpus and size (left). The preference of listeners to assign higher score for signals in compare to others (right).

According to listeners feedback, comparing 11 samples is not an easy task and they had been exhausted. This problem encourages us to estimate the preference of listeners as if they were asked to compare two signals. So the resulting scores from MUSHRA test are used to simulate an AB test. Concretely, each two signals are compared based on their perceptual score. The score values of two signals are converted to a simple comparison in order to simulate the preference of listeners and if the scores are equal, it would be assumed that the listeners does not have any preference. The result is shown on figure 2 (left). The numbers in the heatmap table indicate the preference percentage of

vertical labels against horizontal labels. Based on this figure, the preference of synthetic signals with small voice corpus (left-down) is lower than synthetic signals with large voice corpus (right-up). It confirms that voice corpus reduction decreases the TTS synthetic quality. By looking at cells in large corpus size (left-up), it can be observed that the preference numbers for corpus sizes larger than 15% are around 50. This observation confirms the hypothesis in section 3. It means after a certain voice corpus size the quality of synthetic signal would not be improved perceptually by increasing the voice corpus size.

Figure 2 does not show superiority of spitting greedy in comparison with random strategy. These is contrary to what we expected based on previous studies such as (Chevelu & Lolive, 2015). While it was reported by listeners that the MUSHRA test is not an easy task for this comparison, another perceptual test is proposed for comparing the performance of these two corpus reduction methods.

4.4 The performance of greedy strategy v.s. random selection

Based on listeners' feedback from previous perceptual test, some modifications have been done on listening samples preparation and the platform. While we use the same *test section* and reduction rates, the final listening signals are prepared in a different way. The utterances have been synthesized from the beginning until the first speech pause after 90 diphones. In this way, all samples for sizes/methods will have same content. The duration time of samples are between 5 to 10 seconds. Among 334 utterances of the *test section*, 70 samples have been selected for the listening test according to the highest acoustic distance (Chevelu *et al.*, 2015). The acoustic distance is computed by calculating DTW on the MGC features of two signals. This selection method helps to focus on the most different samples.

An AB test has been prepared with 40 comparison steps. For each step, listeners are asked to give their preference in terms of overall quality between two synthetic signals. These signals have been synthesized by using different voice subcorpora but with same size. Voice subcorpora are a sub part of the *full corpus* selected by the random strategy or the proposed spitting greedy. The reference signal is not provided which lets listeners decide what is the best quality. We hope it makes the task easier. The estimated time for doing the whole test is 15 minutes.

The listening test has been done by 9 listeners. For each voice corpus size between 66 and 70 comparisons have been achieved. Out of 340 comparisons in total, 132 times random strategy has been preferred, 118 times greedy strategy, and 90 times listeners selected no preference. The figure 3 (left) shows the percentage of preference for corpus reduction methods in different voice corpus size.

The figure 3 (left) does not reveal any significant superiority of the modified greedy. Even the synthetic signals for 15% of full voice corpus with random strategy has been evaluated slightly better than modified greedy.

The TTS global costs of the AB test's samples is displayed in the figure 3 (right). This figure shows the synthetic quality of the listening test signals in terms of TTS costs. The TTS global cost given by random selection are not significantly different from those given by the proposed greedy. While the initial problem was synthesizing the rest of the book instead of test section, the the rest of the book has been synthesized by extracted sub-corpus as TTS voice. A same trend as figure 3 is observed for rest of the book (synthetic part). It means that the listening test signals have same synthetic quality as the rest of the book.

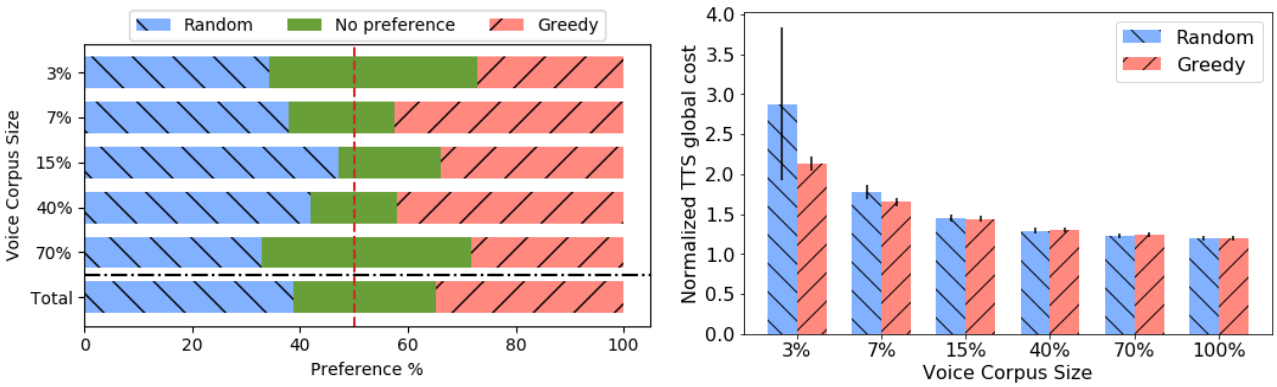


FIGURE 3: The AB test result for different voice corpus size and the total preference of listeners for random and proposed greedy (left). The normalized TTS global cost of listening test samples (right).

It could be concluded that the random reduction works as well as the proposed spitting greedy. The explanation could be the approximation level of the proposed method. It means that reducing the computational time costs a lot in terms of the efficiency of subset solution. Hence the performance of voice subcorpus resulted by proposed greedy becomes close to a random selection.

5 Conclusion

The computational time is the main challenge in voice corpus reduction by greedy algorithm. By modifying the original spitting greedy, its complexity has been reduced to a reasonable time. However this approximation level probably costs lower efficiency and makes the solutions closer to the random strategy.

In the first step, some objective measures like PESQ, DTW between synthetic signal and voice signal, and TTS global cost have been investigated. A perceptual listening test (DMOS) has been designed to evaluate the synthetic signals using different voice subcorpus sizes. A higher correlation between objective measures and perceptual quality confirmed that TTS global cost is the best available metric to estimate perceptual quality.

Hence the TTS global cost has been employed in greedy algorithm for ranking candidates in each reduction step. By modifying the original greedy algorithm, the computational time reduced to a reasonable time. Although these modifications cause some level of inefficiency. The proposed greedy has been compared with random strategy for different voice corpus sizes in a MUSHRA test. It has revealed that after a certain size of voice (1 hour of our audio-book), the voice corpus is big enough and the difference of synthetic signals because of voice corpus size can not be distinguished perceptually. Moreover it has not been observed any difference between random and proposed greedy. In order to evaluate the performance of the proposed greedy, an AB preference test has been run. The result of this listening test confirmed that listeners did not prefer the signals which are synthesized using voice subcorpus obtained with the proposed greedy.

Acknowledgments

Thanks to Damien Lolive, Jonathan Chevelu, and Nelly Barbot for their appreciable help as supervisors. This study has been realized under the ANR (French National Research Agency) project SynPaFlex ANR-15-CE23-0015.

References

ALAIN P., BARBOT N., CHEVELU J., LECORVE G., SIMON C. & TAHON M. (2017). The IRISA text-to-speech system for the blizzard challenge 2017. In *Blizzard Challenge workshop. International Speech Communication Association (ISCA)*, Stockholm, Sweden.

BALJEKAR P. & BLACK A. W. (2016). Utterance selection techniques for TTS systems using found speech. In *9th ISCA Workshop on Speech Synthesis (SSW9)*, p. 184–189, Sunnyvale, USA.

BARBOT N., BOEFFARD O., CHEVELU J. & DELHAY A. (2015). Large linguistic corpus reduction with SCP algorithms. *Computational Linguistics*, **41**(3), 355–383.

BOEFFARD O., CHARONNAT L., LE MAGUER S., LOLIVE D. & VIDAL G. (2012). Towards fully automatic annotation of audio books for tts. In *Eighth International Conference on Language Resources and Evaluation (LREC)*, p. 975–980, Istanbul, Turkey.

BRAUNSCHWEILER N. & BUCHHOLZ S. (2011). Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality. In *Twelfth Annual Conference of the International Speech Communication Association (InterSpeech)*, p. 1821–1824, Florence, Italy.

CHEVELU J. & LOLIVE D. (2015). Do not build your TTS training corpus randomly. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, p. 350–354, Nice, France : IEEE.

CHEVELU J., LOLIVE D., MAGUER S. L. & GUENNEC D. (2015). How to compare TTS systems : a new subjective evaluation methodology focused on differences. In *Sixteenth Annual Conference of the International Speech Communication Association (InterSpeech)*, p. 3481–3485, Dresden, Germany.

CHU M. & PENG H. (2001). An objective measure for estimating MOS of synthesized speech. In *Seventh European Conference on Speech Communication and Technology (EuroSpeech)*, p. 2087–2090, Aalborg, Denmark.

COOPER E., CHANG A., LEVITAN Y. & HIRSCHBERG J. (2016). Data selection and adaptation for naturalness in hmm-based speech synthesis. In *Seventeenth Annual Conference of the International Speech Communication Association (InterSpeech)*, p. 357–361, San Francisco, USA.

ESPINOSA D., WHITE M., FOSLER-LUSSIER E. & BREW C. (2010). Machine learning for text selection with expressive unit-selection voices. In *Eleventh Annual Conference of the International Speech Communication Association (InterSpeech)*, p. 1125–1128, Makuhari, Japan.

FRANÇOIS H. & BOEFFARD O. (2002). The greedy algorithm and its application to the construction of a continuous speech database. In *Third International Conference on Language Resources and Evaluation (LREC)*, volume 5, p. 1420–1426, Las Palmas, Spain.

FREEDMAN D., PISANI R. & PURVES R. (2007). Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

KENDALL M. G. (1948). Rank correlation methods.

KRUL A., DAMNATI G., YVON F., BOIDIN C. & MOUDENC T. (2007). Approaches for adaptive database reduction for text-to-speech synthesis. In *Eighth Annual Conference of the International Speech Communication Association (InterSpeech)*, p. 2881–2884, Antwerp, Belgium.

RECOMMENDATION I. (2003). 1534-1 : Method for the subjective assessment of intermediate quality level of coding systems. *International Telecommunication Union*.

RIX A. W., BEERENDS J. G., HOLLIER M. P. & HEKSTRA A. P. (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, p. 749–752, Salt Lake City, USA : IEEE.

SHAMSI M., CHEVELU J., LOLIVE D. & BARBOT N. (2020). Corpus design for expressive speech : impact of the utterance length. In *International Conference of Speech Prosody*.

SHAMSI M., LOLIVE D., BARBOT N. & CHEVELU J. (2019). Corpus design using convolutional auto-encoder embeddings for audio-book synthesis. In *Twentieth Annual Conference of the International Speech Communication Association (InterSpeech)*, p. 1531–1535, Graz, Austria.

SPEARMAN C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, **15**(1), 72–101.

TODA T., KAWAI H., TSUZAKI M. & SHIKANO K. (2006). An evaluation of cost functions sensitively capturing local degradation of naturalness for segment selection in concatenative speech synthesis. *Speech Communication*, **48**(1), 45–56.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Exploiter des modèles de langue pour évaluer des sorties de logiciels d'OCR pour des documents français du XVII^{e} siècle

Jean-Baptiste Tanguy
CELLF, STIH, Sorbonne Université, 1 rue Victor Cousin, 75005, Paris, France
jean-baptiste.tanguy@sorbonne-universite.fr

RÉSUMÉ

Pour comparer deux sorties de logiciels d'OCR, le *Character Error Rate* (ou, *CER*) est fréquemment utilisé. Moyennant l'existence d'une vérité de terrain de qualité pour certains documents du corpus, le *CER* calcule le taux d'erreur de ces pièces et permet ensuite de sélectionner le logiciel d'OCR le plus adapté. Toutefois, ces vérités de terrain sont très coûteuses à produire et peuvent freiner certaines études, même prospectives. Nous explorons l'exploitation des modèles de langue en agrégeant selon différentes méthodes les probabilités offertes par ceux-ci pour estimer la qualité d'une sortie d'OCR. L'indice de corrélation de Pearson est ici utilisé pour comprendre dans quelle mesure ces estimations issues de modèles de langue covarient avec le *CER*, mesure de référence.

ABSTRACT

Language Model Based Evaluation of OCR Software Output Qualities for 17th Century French Documents

In order to compare two OCR software outputs, the *Character Error Rate* (or, *CER*) is frequently used. When a quality ground truth exists for several documents, the *CER* calculates the error rate for these documents and therefore allows to choose the most suitable OCR software. However, these ground truths are extremely expensive and may slow down some studies. Hence, we are exploring the exploitation of language models by the agregation (with several methods) of their probabilities in order to estimate OCR output qualities. The Pearson correlation is used to understand how these language model based estimations covary with the *CER*, reference metric.

MOTS-CLÉS : OCR, modèle de langue, évaluation, document historique, français pré-classique.

KEYWORDS : OCR, language model, evaluation, historical document, pre-classical French.

1 Introduction

Les campagnes de numérisation des collections patrimoniales s'installent à la frontière de deux enjeux relatifs aux documents historiques : leur pérennisation et leur accessibilité. La Bibliothèque Nationale de France, qui a commencé la numérisation de ses fonds au début des années 1990 (avec l'arrivée de Gallica en 1997 (Bermes, 2020)), et la Bibliothèque Mazarine, qui a engagé en 2014 la numérisation de sa collection d'incunables et en 2015 celle de ses Mazarinades [1], sont ici exemplaires. Au-delà de la construction d'éditions web, la numérisation de telles collections rend possible leur exploitation automatique à grande échelle, moyennant une transcription de leur contenu textuel. Ceci

1. Documents parus en France, lors de la Fronde (1648-1653).

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 205–217.
HAL : hal-02786201.

constitue un réel intérêt, tant pour la communauté savante que pour le grand public. Toutefois, deux problèmes majeurs se posent. D'une part, les logiciels de reconnaissance optique de caractères (ou OCR), s'ils offrent des transcriptions automatiques de qualité pour des documents contemporains générés électroniquement, sont nettement moins robustes face à des documents historiques. (Lejeune & Abiven, 2019), pour le « corpus » des Mazarinades, exposent un ensemble d'éléments rendant l'étude de ces documents historiques particulièrement complexe : variantes graphiques, abréviations, orthographe erratique mais aussi un « état inégal de conservation [des] imprimés souvent produits dans l'urgence et l'économie de moyens (papier et encre de mauvaise qualité, notamment) ». D'autre part, et s'agissant de processus automatisés, la connaissance de la qualité des sorties des logiciels d'OCR est primordiale. Néanmoins, l'évaluation d'outils d'OCR n'est pas stable d'un corpus à l'autre, car elle fait intervenir des corpus particulièrement hétérogènes (Springmann *et al.*, 2014). Mesurer la qualité d'une sortie d'OCR nécessite alors, au moins pour un ensemble réduit de la collection à numériser et à océriser, une transcription manuelle et certaine à laquelle les sorties d'OCR seront comparées ; et ce, dès lors qu'une nouvelle collection est à océriser. Or cette transcription de référence, qu'on appelle vérité de terrain (Springmann *et al.*, 2018), est coûteuse à constituer ce qui limite d'autant la quantité de données disponible pour l'évaluation. Ainsi, estimer la qualité des sorties de logiciels d'OCR sans vérité de terrain permettrait d'opter à moindre coût pour un logiciel d'OCR adapté. Il s'agit donc d'une démarche d'évaluation non supervisée.

Dans cet article, nous proposons i) d'apprendre des modèles de langue sur un corpus en français pré-classique (XVII^{e} siècle), ii) de parcourir des sorties de logiciels d'OCR par fenêtre glissante en récupérant les probabilités de chaque modèle de langue de rencontrer une telle séquence de caractères pour enfin iii) estimer la qualité de ces sorties d'OCR. Différentes méthodes d'agrégation des probabilités précitées sont proposées pour estimer la qualité globale des *pages* océrisées. L'étude des corrélations entre ces estimations et les *CER* [2] (pour chaque page du corpus) permettra de valider ou réfuter la pertinence de ces estimateurs sur le corpus de l'étude.

Après l'exposition de plusieurs méthodologies d'estimation non supervisée de la qualité de sorties d'OCR (section 2), nous décrivons le cadre expérimental de notre étude, où le corpus, les modèles de langue et les méthodes d'agrégation de leurs probabilités sont décrits (section 3). Nous présenterons finalement les résultats de l'expérimentation en section 4.

2 Estimer la qualité de sorties de logiciels d'OCR

Pour évaluer des sorties d'OCR en échappant à la fastidieuse transcription des imprimés, plusieurs méthodes conduisent à la création de nouvelles mesures d'évaluation. Celles-ci sont comparées à des mesures de référence (le *CER* ou la précision), calculées grâce à des vérités de terrain, pour valider ou réfuter leur pertinence.

Exploiter des ressources lexicales (Springmann *et al.*, 2016) proposent d'estimer la qualité d'une sortie d'OCR en exploitant la *lexicalité* de celle-ci. La *lexicalité* est calculée en faisant la moyenne, pour chaque *token* observé dans la sortie d'OCR, des distances de Levenshtein entre ces *tokens* et leur supposé équivalent moderne le plus proche (*supposé* car la relation entre deux formes de deux états différents d'une même langue n'est pas nécessairement bijective). Il est montré que cette *lexicalité*

2. *Character Error Rate*.

est très clairement corrélée à la précision.

Exploiter les valeurs de confiance des logiciels d'OCR Une autre voie empruntée par (Springmann *et al.*, 2016) est de mettre à profit les valeurs de confiance des logiciels d'OCR. Ceux-ci renvoient en effet une valeur correspondant à l'intensité de la confiance que le logiciel associe au caractère qu'il propose. Dans le cas d'une hésitation entre deux caractères proches (par exemple, *G* et *O*), le conflit est traduit par deux valeurs de confiance similaires et plus faibles que dans le cas d'une certitude pour un caractère en particulier. Les auteurs supposent que « la somme des valeurs de confiance associées aux caractères de sortie doit ainsi être corrélée avec la précision de la sortie d'OCR » [3] et le vérifient très nettement.

Exploiter les *bounding boxes* Avant de proposer un ensemble de caractères, les logiciels d'OCR segmentent les images proposées à l'océrisation. Ces segmentations (en colonnes, en lignes, en mots ou encore en caractères) apparaissent pour (Gupta *et al.*, 2015) comme de bons indicateurs pour estimer la qualité d'une sortie d'OCR. En effet, s'agissant d'un processus en cascade, si la segmentation fait défaut, l'océrisation en pâtira largement. En recueillant les informations graphiques associées aux objets résultant de la segmentation (les *bounding boxes*), les auteurs proposent d'apprendre un modèle de classification permettant de distinguer deux types de *bounding boxes* : les *bounding boxes* pertinentes (*BBs*) et les *bounding boxes* non pertinentes (*noise BBs*) [4]. Le calcul de cet estimateur se réalise ensuite en comptant la proportion des *noise BBs*. Il est conclu que cette proportion de *noise BBs* permet d'estimer de manière satisfaisante la qualité globale d'un document océrisé mais aussi d'en identifier les passages bruités.

Exploiter les modèles de langue Les modèles de langue, appris au grain mot, sont fréquemment utilisés en reconnaissance de la parole. (Chen *et al.*, 1998) ont proposé d'utiliser les modèles de langue non pas pour corriger en post-traitement les sorties de reconnaissance d'un flux de parole mais pour estimer la qualité de cette sortie. La perplexité et ses dérivés (comme la log-perplexité) y apparaissent fortement corrélées au *word error rate* (le taux d'erreur mot, ou, *WER*) avec, pour le premier jeu de données de leur étude, une relation presque parfaitement linéaire. Néanmoins, pour l'évaluation de sorties d'OCR, les modèles de langue ne semblent pas avoir encore été testés.

Utiliser des pseudo-vérités de terrain (Ul-Hasan *et al.*, 2016) proposent d'utiliser la sortie d'un logiciel d'OCR (en l'occurrence, `Tesseract`) comme une pseudo-vérité de terrain sur laquelle est appris un premier modèle. Si l'objectif de ce travail n'est pas d'estimer la qualité d'une sortie d'OCR, les auteurs se soucient du manque de transcriptions à disposition et atteignent avec ces pseudo-vérités de terrain des précisions de l'ordre de 95% sur des documents imprimés du XVII^{e} siècle.

Comme (Chen *et al.*, 1998), nous proposons d'utiliser des modèles de langue (mais appris au grain caractère) pour évaluer les sorties des logiciels d'OCR. Il s'agit de récupérer une probabilité pour chaque caractère (voir sous-section 3.4), d'agréger ces probabilités et d'observer, comme (Springmann *et al.*, 2016), s'il existe une corrélation entre ces agrégations et la mesure de référence *CER*.

3. *The sum of the confidences over all output characters should therefore correlate with the accuracy of the output.*
4. Par exemple, une *bounding boxe* pertinente encadre une ligne ou un mot alors qu'une *bounding boxe* non pertinente encadre deux lignes juxtaposées mais appartenant à deux colonnes différentes.

3 Cadre expérimental

Dans cette section, nous décrivons le corpus de notre étude, la méthodologie et les logiciels d'océrisation, les modèles de langue appris et les mesures d'évaluation à comparer au *CER*.

3.1 Un corpus d'œuvres françaises du XVII^{e} siècle

Titre	Auteur	Date	Domaine	Nb pages	Nb lignes	Nb mots
Oraisons funebres	Bossuet	1683	Théologie	27	770	4 128
La Pucelle...	Chapelain	1656	Poésie	28	753	4 735
Advis sur la peste	Ellain	1606	Science	22	618	3 168
Egalite des hommes et des femmes	Gournay	1622	Philosophie	31	825	4 284
La Maniere d'amolir les os...	Papin	1682	Science	23	548	2 230
Experiences Nouvelles...	Pascal	1647	Science	39	776	3 568
Introduction à la vie devote	Sales	1641	Théologie	25	618	3 915
Oeuvres completes (Tome II.)	Viau	1623	Poésie	33	852	4 055

TABLE 1 – Description des œuvres du corpus.

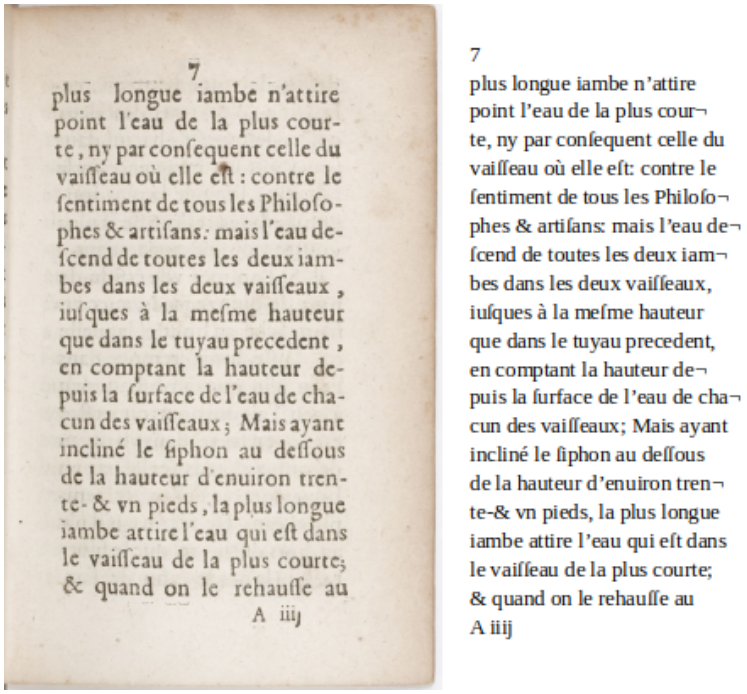


FIGURE 1 – Numérisation de la page 15 des *Experiences Nouvelles touchant le vide...* de Pascal (1647) présentée avec sa transcription diplomatique.

Rassemblé et transcrit par (Gabay, 2019), le corpus de travail est une sélection de certaines œuvres françaises du XVII^{e} siècle décrites dans la table 1. Notre corpus environ 6 000 lignes, 30 000 mots et 15 0000 caractères. Un exemple de numérisation est proposé dans la figure 1.

i)				ii)			
Identifiant	**Nb lignes**	**Nb mots**	**Nb caractères**	**Identifiant**	**Nb lignes**	**Nb mots**	**Nb caractères**
Bossuet-1683	27	770	4 128	Papin-1682	23	548	2 230
Chapelain-1656	28	753	4 735	Pascal-1647	39	776	3 568
Ellain-1606	22	618	3 168	Sales-1641	25	618	3 915
Gournay-1622	31	825	4 284	Viau-1623	33	852	4 055

TABLE 2 – Description des sous-corpus dédiés à i) l'apprentissage des modèles de langue et ii) l'océrisation et l'évaluation de la qualité des sorties d'OCR.

Les variétés thématique et diachronique du corpus ainsi que les transcriptions diplomatiques de grande qualité permettent de le considérer non seulement comme un premier laboratoire privilégié pour l'étude de l'OCR mais aussi comme représentant de cet état de langue. Ainsi les œuvres de Bossuet, Chapelin, Ellain et Gournay et les œuvres de Papin, Pascal, Sales et Viau constituent-elles deux sous-corpus : les transcriptions des premières permettant l'apprentissage des modèles de langues et les images et les transcriptions des secondes l'application des logiciels d'OCR et la mesure du *CER* respectivement (voir la table 2).

3.2 Océrisation

Le corpus dédié à l'océrisation est composé de 120 pages numérisées, toutes avec une résolution de 400dpi. Afin de réaliser l'océrisation de ces images, deux logiciels ont été utilisés :

— `Kraken`, version 2.0.8 [5] (voir (Kiessling, 2019)) ;
— `Tesseract`, version 0.3.3 [6] (voir (Smith, 2007)).

Pré-traitement des images Pour appliquer ses modèles de reconnaissance de caractères, `Kraken` prend en entrée des *lignes* binarisées (un pixel ne peut être que blanc ou noir) alors que `Tesseract` peut admettre des pages entières en nuances de gris ou même en couleurs. Dans un souci d'unité, et puisque `Kraken` est plus restrictif que `Tesseract`, toutes les images ont été segmentées et binarisées en utilisant les modules dédiés de `Kraken`. Ceci constitue un biais au regard des performances de `Tesseract` ; néanmoins, et selon notre hypothèse, le taux d'erreur devrait évoluer dans le même sens que les métriques d'estimation de l'étude.

Application des modèles L'objectif étant d'observer dans quelle mesure les modèles de langue peuvent être de bons indicateurs de la qualité d'une sortie d'OCR, l'utilisation de plusieurs modèles, adaptés ou non aux documents de l'étude, apparaît primordiale. Pour ce faire, les modèles de `Kraken` (anglais contemporain) et de `Tesseract` (français contemporain) ont été appliqués aux lignes segmentées et binarisées ainsi qu'un modèle `Kraken` appris sur ces mêmes données [7]. On dispose alors *a priori* de deux modèles non adaptés aux documents de l'étude [8] et d'un modèle suradapté à ces documents (puisqu'appris sur ceux-ci). L'hypothèse que nous faisons est que l'agrégation des proba-

5. http://kraken.re/ et https://pypi.org/project/kraken/
6. https://pypi.org/project/pytesseract/
7. https://github.com/e-ditiones/OCR17
8. Par exemple, le <ſ> ne fait pas partie du vocabulaire des modèles de `Kraken` (anglais) et de `Tesseract` (français).

bilités offertes par les modèles de langue sur les sorties des modèles de `Kraken` et de `Tesseract` sera plus faible que sur les sorties du modèle `Kraken` appris sur ces mêmes données.

Lignes Kraken	CER	Lignes Tesseract	CER	Lignes Kraken 17	CER
	100,0 %		100,0 %	¬	100,0 %
plus longue iambe n'attire	3,8 %	plus Jongue iambe n'attire	7,6 %	plus longue iambe n'attire	0 %
point lcau de la plus cour-	7,4 %	point l'eau de la plus cour-	3,7 %	point l'eau de la plus cour¬	0 %
te, ny par confequent celle du	3,4 %	te, ny par confequent celle du	3,4 %	te, ny par confequent celle du	0 %
vaifeau oi elle ef : contre le	16,6 %	vaiffeau où elle et : contre le	10 %	vaiſſeau ou elle elt : contre le	6,6 %
fentiment de tous les Philofo-	6,8 %	fentiment de tous les Philofo-	6,8 %	ſentiment de tous les Philoſo¬	0 %
phes artifans : maislcau de-	14,8 %	phes & artifans : mais l'eau de-	7,4 %	phes & artiſans. mais l'eau de¬	0 %
fcend de toutes lcs dcuxiam-	14,2 %	fcend de toutes les deuxiam-	7,1 %	ſcend de toutes les deux iam¬	0 %
bes dans les dcux vaiffeaux,	11,1 %	bes dans les deux vaiffeaux ,	7,4 %	bes dans les deux vaiſſeaux,	0 %
iufques a la mefme hauteur	11,5%	iufques à la mefme hauteur	7,6 %	iuſques à la meſme hauteur	3,8 %
que dans le tuyau preccdent,	3,7 %	que dans le tuyau precedent ,	0 %	que dans le tuyau precedent,	0 %
en comptant la hautcur dec-	13,6 %	en comptant la hauteur de-	4,5%	en comptant la hauteur de¬	0 %
puis la furface deleau de cha-	12,9 %	puis la furface de l'eau de cha-	6,4 %	puis la furface de l'eau de cha¬	0 %
cun des vaiffcaux ; Mais ayant	10,7 %	cun des vaiffeaux ; Maisayant	10,7 %	cun des vaiſſeaux ; Mais ayant	0 %
inclin lc fiphon au deffous	17,8 %	incliné le fiphon au deffous	10,7 %	incliné le ſiphon au deſſous	3,7 %
dc la hautcur dcnuiron trcn-	17,8 %	de la hauteur d'enuiron tren-	3,5 %	de la hauteur d'enuiron tren¬	0 %
te- vn picds, lapluslongue	11,5 %	te- & vn picds, la plus longue	3,8 %	te & vn pieds, la plus longue	0 %
iambe attirclcau qui eft dans	16,1 %	jambe attire l'eau qui eft dans	9,6 %	iambe attire l'eau qui eſt dans	0 %
le vaifeau de la plus courte ;	6,8 %	le vaiffeau de la plus courtc ;	10,3 %	e vaiſſeau de la plus courte ;	3,4 %
quand on le rehaufe au	8,6 %	& quand on le rehaufle au	8,6 %	& quand on le rehauſſe au	0 %
A iii	16,6 %	À ill	66,6 %	A iii	16,6 %

TABLE 3 – Sorties des trois modèles d'OCR pour la 15e page des *Experiences Nouvelles touchant le vide…* de Pascal (1647). De gauche à droite : modèle `Kraken` (anglais), modèle `Tesseract` (français) et modèle `Kraken` (français du XVIIe siècle).

La table 3 présente un exemple d'utilisation des trois modèles d'OCR sélectionnés. Les lignes affichées sont les sorties des modèles et les *CER* ont été calculés face à la référence (présentée dans la figure 1).

3.3 Apprentissage des modèles de langue

Deux types de modèles de langue (au grain caractère) ont été appris sur le sous-corpus dédié (voir la table 2) qui compte 121 caractères différents :

— des modèles de langue à probabilités conditionnelles, appris comme la probabilité d'observer un caractère sachant une séquence de caractères (un historique) ;
— des modèles de langue appris par des réseaux de neurones (LSTM et biLSTM).

Le premier type de modèles constitue une *baseline* puisque ces modèles sont simplement construits en comptant, par fenêtre glissante sur le corpus d'apprentissage, le nombre d'occurrences du caractère suivant la séquence de caractères contenue dans la fenêtre glissante. Ces occurrences absolues sont ensuite divisées par la somme des occurrences de tous les caractères suivants cette séquence et sont utilisées comme des probabilités, puisque contenues dans l'intervalle $[0; 1]$.

Les modèles de langue LSTM et biLSTM ont été choisis pour confronter aux modèles de la *baseline* des modèles appris par réseaux de neurones, en l'occurrence des réseaux de neurones récurrents. Ces modèles ont été appris en utilisant les modèles séquentiels de la librairie Python *keras*. Un *mapping* du vocabulaire est d'abord réalisé en prétraitement [9]. Les réseaux LSTM et biLSTM contiennent tous une couche *LSTM* et aux réseaux biLSTM est ajoutée une couche *Bidirectional* ; l'hypothèse étant ici

9. À chaque élément du vocabulaire (entendu comme l'ensemble des caractères différents) est associé un entier dans une table.

que i) tout caractère ne peut suivre tout autre caractère et que ii) tout caractère ne peut être précédé de tout autre caractère. Enfin, la fonction *softmax* est utilisée comme fonction d'activation. Ces modèles de langue ont été appris sur des séquences de *n* caractères, pour *n* variant de 2 à 10 et le nombre d'époques pour chaque apprentissage est 100. Finalement, on dispose donc de $3 * 9 = 27$ modèles de langue pour tester l'estimation de la qualité des sorties des logiciels d'OCR. Le nombre de caractères dans le vocabulaire de ces modèles de langue est de 121.

3.4 Métriques d'estimation

3.4.1 Préambules

Calcul du *CER* Le *CER* est calculé entre une suite de caractères de référence et une suite à tester comme la somme des insertions, délétions et substitutions divisée par le nombre total de caractères de la chaîne de référence. Il peut être supérieur à 1 si le nombre d'insertions est particulièrement élevé.

Probabilités des modèles de langue Les modèles de langue permettent de disposer de la probabilité qu'un caractère donné suive une certaine séquence de caractères. Si une sortie de logiciel d'OCR est parcourue par une fenêtre glissante à partir de laquelle est renvoyée une séquence de caractères et le caractère suivant cette séquence, pour une sortie d'OCR on dispose d'une suite d'au plus $C - n$ probabilités, avec C le nombre total de caractères et n la taille de la fenêtre glissante en caractères. *Au plus* car il est possible que certains caractères fournis par le modèle d'OCR n'aient pas été rencontrés dans le corpus d'apprentissage du modèle de langue [10].

On cherche donc à agréger ces probabilités, pour chaque document du corpus océrisé, dans l'objectif que ces agrégats soient corrélés au *CER* qu'on peut calculer grâce aux transcriptions. Il s'agit de calculer d'autres métriques ne nécessitant pas de vérité de terrain (à partir des probabilités fournies par les modèles de langue) et de valider ou réfuter la pertinence de leur estimation de la qualité d'une sortie d'OCR face à une métrique de référence, le *CER*.

3.4.2 Agrégations des probabilités des modèles de langue

La somme des probabilités Une première métrique peut être la somme des probabilités renvoyées par les modèles de langue. Sous réserve que les modèles de langue sont bien des distributions de probabilités, la somme des probabilités d'une suite de caractères correspondant à du texte est de 1 alors qu'elle ne peut l'être dans le cas contraire. Ainsi, pour une sortie d'OCR, on a :

$$S = \sum_{i=n+1}^{C-n} P_{LM}(c_i|h_{n,i})$$

Avec P_{LM} la probabilité renvoyée par un modèle de langue LM, n la taille de la fenêtre glissante en caractères, C le nombre total de caractères de la sortie d'OCR, c_i le i^{e} caractère de la sortie d'OCR et $h_{n,i}$ l'historique de n caractères du caractère c_i.

10. Par exemple, `Kraken` (anglais) et `Tesseract` (français), appris sur des documents contemporains, ont dans leur vocabulaire le symbole € et peuvent le proposer dans leur océrisation. Pour un modèle de langue appris sur des données textuelles françaises du XVII$^{\text{e}}$ siècle, ce symbole n'existe pas.

Le produit des probabilités Le produit des probabilités peut aussi constituer une autre métrique d'estimation. Il rend compte de la probabilité d'une suite de caractères selon l'hypothèse, ici réductrice, de l'indépendance. Il est défini comme :

$$Pr = \prod_{i=n+1}^{C-n} P_{LM}(c_i|h_{n,i})$$

La perplexité Plus couramment utilisée pour juger de la qualité d'un modèle de langue, la perplexité est la probabilité inverse de la sortie d'OCR normalisée par son nombre de caractères. Puisqu'elle mesure la distance entre la fonction de probabilité et les données de l'ensemble de test, elle semble pertinente à tester comme métrique d'estimation. Elle est définie comme :

$$PP = \frac{1}{(\prod_{i=n+1}^{C-n} P_{LM}(c_i|h_{n,i}))^{\frac{1}{C-n}}}$$

La log-perplexité Enfin, la log-perplexité peut aussi constituer une métrique d'estimation ; (Chen *et al.*, 1998) montrent qu'elle aussi est corrélée au *WER* dans le domaine de la reconnaissance de la parole.

3.4.3 Échelles de calcul des agrégations

Les agrégations des probabilités précitées peuvent être calculées à plusieurs échelles : celle de l'œuvre, de la page, de la ligne ou encore du mot. Puisque la perplexité PP est calculée comme l'inverse d'une racine n-ième, elle tend vers 1 à mesure que le nombre total de caractères $C-n$ grandit. Plus le nombre de caractères sur lesquels elle est calculée grandit, moins elle est informative. Les estimateurs de qualité d'océrisation des pages sont donc calculés comme la moyenne des agrégations des probabilités calculées à l'échelle du mot [11].

Notons que réaliser une *moyenne* constitue un biais important. Les données textuelles issues d'OCR n'ont pas une qualité homogène pour une même œuvre ou une même page ; la moyenne efface ces disparités pourtant essentielles à soulever. D'autre part, certains mots comportent trop peu de caractères pour que le modèle de langue puisse leur calculer une probabilité. Certains passages sont ignorés et la moyenne ne le traduit pas.

4 Expérimentations et résultats

Pour calculer les métriques d'estimation de la qualité de l'OCR sur une page du corpus, on calcule ces métriques pour chaque mot de la page et on en fait la moyenne. Afin de confirmer ou réfuter l'intérêt de ces métriques, un calcul de corrélation Pearson [12] est réalisé avec le *CER*. Si une ou

11. La tokenisation est réalisée par une simple segmentation par l'espace des chaînes de caractères.

12. En tant que normalisation des covariances, le coefficient de corrélation exprime à quel point deux variables sont liées. Ce coefficient étant une normalisation, il appartient à l'intervalle $[-1;1]$; les corrélations positives indiquent que les deux variables évoluent dans le même sens et les corrélations négatives qu'elles évoluent dans un sens opposé. Plus une corrélation est proche de 1 ou -1, plus le lien entre les deux variables est fort ; au contraire, plus la corrélation est proche de 0, plus ce lien se dissipe.

plusieurs métriques est corrélée-s significativement au *CER*, on peut conclure que l'apprentissage d'un modèle de langue sur des données du français du XVII^{e} siècle et l'utilisation de ses probabilités pour estimer la qualité d'une sortie d'OCR sur des documents de la même période sont justifiés.

4.1 Préambule à l'analyse des corrélations

Métriques	**Variations**	**Signes des corrélations avec le *CER***
CER	↗	
S	↘	-
Pr	↘	-
$Pr^{\frac{1}{C-n}}$	↘	-
$PP = \frac{1}{Pr^{\frac{1}{C-n}}}$	↗	+
$\log(PP)$	↗	+

TABLE 4 – Variations et signes des corrélations avec le *CER* des métriques d'estimation pour un nombre d'erreurs qui augmente.

La table 4 expose la variation des métriques d'estimation et le signe de leur corrélation avec le *CER* pour un nombre d'erreurs d'océrisation qui augmente. L'hypothèse est que les modèles de langue fournissent des probabilités (voir le paragraphe 3.4.1) plus élevées face à une sortie d'OCR sans erreur (du *texte*) et des probabilités plus faibles face à une sortie d'OCR avec erreurs (du *non-texte*). Ainsi, pour valider les métriques comme estimateurs pertinents, les corrélations entre le *CER* et la somme et le produit des probabilités doivent être négatives alors qu'elles doivent être positives entre le *CER* et la perplexité et la log-perplexité.

4.2 Corrélations entre le *CER* et les métriques d'estimation

Le sous-corpus dédié à l'OCR est composé de 118 pages. On peut donc, pour les sorties des trois modèles d'OCR utilisés et pour les trois types de modèles de langue, calculer les métriques d'estimation et le *CER*, et ce pour chaque page du corpus.

OCR : Kraken (français contemporain). ML : probabilités conditionnelles.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,063	0,496	0,111	0,226	-0,016	0,862	-0,013	0,884
n=3	-0,098	0,287	0,073	0,426	0,005	0,955	0,021	0,820
n=4	-0,073	0,428	-0,043	0,642	-0,010	0,913	-0,014	0,879
n=5	-0,137	0,135	-0,043	0,638	-0,015	0,868	-0,026	0,780
n=6	-0,093	0,314	0,000	0,996	0,067	0,466	0,059	0,522
n=7	-0,035	0,708	-0,032	0,728	0,130	0,157	0,117	0,205
n=8	-0,064	0,485	-0,074	0,420	0,043	0,643	0,054	0,560
n=9	-0,057	0,538	-0,012	0,898	0,018	0,846	0,021	0,821
n=10	-0,046	0,615	-0,023	0,806	0,024	0,794	0,026	0,780

OCR : Tesseract (anglais contemporain). ML : probabilités conditionnelles.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,004	0,968	0,158	**0,086**	0,113	0,221	0,006	0,952

n=3	-0,130	0,156	0,009	0,920	-0,003	0,976	0,056	0,540
n=4	-0,124	0,178	-0,005	0,960	0,016	0,866	0,060	0,518
n=5	-0,158	**0,084**	-0,070	0,449	0,134	0,143	0,158	**0,085**
n=6	-0,138	0,133	-0,054	0,556	0,180	**0,049**	0,188	**0,040**
n=7	-0,100	0,278	-0,027	0,773	0,093	0,313	0,084	0,359
n=8	-0,055	0,547	-0,008	0,930	-0,006	0,949	-0,008	0,928
n=9	-0,054	0,554	-0,083	0,366	0,096	0,299	0,095	0,300
n=10	-0,024	0,796	-0,212	**0,020**	0,228	**0,012**	0,187	**0,041**

OCR : Kraken (français XVII^{e}). ML : probabilités conditionnelles.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,052	0,572	0,129	0,162	0,238	**0,009**	0,040	0,663
n=3	-0,080	0,384	0,087	0,343	-0,003	0,970	0,030	0,742
n=4	-0,041	0,654	-0,010	0,916	0,012	0,894	0,031	0,739
n=5	-0,111	0,225	-0,072	0,437	0,039	0,670	0,069	0,452
n=6	-0,135	0,142	-0,055	0,551	0,138	0,133	0,143	0,120
n=7	-0,086	0,348	-0,030	0,746	0,063	0,496	0,075	0,414
n=8	-0,096	0,296	-0,030	0,743	-0,045	0,622	-0,045	0,625
n=9	-0,052	0,574	-0,017	0,857	-0,015	0,874	-0,029	0,757
n=10	-0,021	0,817	-0,039	0,669	0,121	0,189	0,097	0,291

OCR : Kraken (français contemporain). ML : LSTM.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,059	0,520	0,094	0,305	0,046	0,615	0,000	0,999
n=3	-0,078	0,395	0,078	0,397	-0,031	0,739	-0,011	0,902
n=4	-0,094	0,306	-0,070	0,446	0,113	0,221	-0,111	0,227
n=5	-0,108	0,240	-0,048	0,600	-0,061	0,511	-0,077	0,404
n=6	-0,039	0,675	0,184	0,045	0,055	0,552	-0,092	0,320
n=7	0,198	**0,030**	-0,049	0,595	-0,051	0,578	0,035	0,702
n=8	-0,034	0,709	0,026	0,781	-0,017	0,854	0,058	0,530
n=9	-0,055	0,549	-0,016	0,860	-0,062	0,499	0,076	0,407
n=10	0,063	0,492	-0,036	0,695	-0,055	0,550	-0,061	0,506

OCR : Tesseract (anglais contemporain). ML : LSTM.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,010	0,911	0,138	0,131	-0,032	0,730	-0,081	0,377
n=3	-0,132	0,151	-0,046	0,621	-0,004	0,962	0,133	0,148
n=4	-0,085	0,354	-0,053	0,567	0,003	0,972	-0,009	0,926
n=5	-0,096	0,298	-0,080	0,383	-0,022	0,808	0,053	0,568
n=6	-0,107	0,245	0,034	0,716	-0,010	0,915	-0,038	0,683
n=7	-0,116	0,208	-0,079	0,390	0,106	0,250	0,007	0,938
n=8	0,043	0,640	-0,026	0,776	-0,034	0,711	-0,025	0,788
n=9	-0,044	0,636	0,024	0,791	-0,032	0,732	0,029	0,756
n=10	-0,025	0,788	-0,072	0,432	0,012	0,895	0,073	0,426

OCR : Kraken (français XVII^{e}). ML : LSTM.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,055	0,552	0,168	**0,066**	-0,006	0,944	-0,049	0,596
n=3	-0,065	0,482	0,114	0,215	-0,023	0,804	-0,062	0,503
n=4	-0,058	0,526	-0,030	0,741	-0,026	0,777	-0,024	0,793
n=5	-0,069	0,455	-0,052	0,576	-0,012	0,898	-0,005	0,953
n=6	-0,083	0,366	0,045	0,623	-0,008	0,930	0,017	0,856
n=7	-0,067	0,465	-0,049	0,595	-0,027	0,773	-0,056	0,541
n=8	-0,104	0,258	-0,041	0,653	-0,030	0,745	-0,029	0,751
n=9	-0,023	0,805	-0,036	0,697	-0,022	0,809	-0,034	0,710
n=10	0,181	**0,048**	-0,059	0,520	-0,006	0,947	-0,012	0,898

OCR : Kraken (français contemporain). ML : biLSTM.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,042	0,645	0,115	0,213	-0,040	0,661	-0,043	0,641
n=3	-0,098	0,289	0,160	**0,081**	-0,096	0,295	-0,077	0,404
n=4	-0,091	0,320	-0,087	0,346	0,085	0,357	-0,126	0,170

n=5	-0,019	0,837	-0,085	0,354	-0,049	0,595	-0,013	0,891
n=6	-0,076	0,411	0,131	0,154	-0,040	0,662	-0,087	0,345
n=7	0,010	0,914	-0,105	0,255	-0,058	0,529	0,009	0,925
n=8	-0,053	0,564	-0,085	0,357	0,623	**0,001**	0,053	0,563
n=9	-0,070	0,446	-0,060	0,517	-0,024	0,794	-0,054	0,560
n=10	0,084	0,361	-0,028	0,758	-0,033	0,722	-0,059	0,521

OCR : Tesseract (anglais contemporain). ML : biLSTM.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	0,013	0,886	0,184	**0,044**	0,000	1,000	-0,117	0,203
n=3	-0,137	0,137	-0,022	0,816	0,117	0,203	0,157	**0,087**
n=4	-0,040	0,663	0,011	0,909	-0,025	0,788	-0,051	0,577
n=5	-0,079	0,389	-0,055	0,547	-0,006	0,946	0,023	0,800
n=6	-0,058	0,530	-0,012	0,893	-0,045	0,627	-0,098	0,287
n=7	-0,064	0,485	-0,013	0,885	-0,007	0,943	-0,093	0,314
n=8	0,036	0,696	-0,023	0,801	0,051	0,580	-0,036	0,694
n=9	-0,053	0,566	-0,033	0,724	0,007	0,936	0,050	0,586
n=10	-0,029	0,754	0,029	0,756	-0,024	0,797	0,027	0,773

OCR : Kraken (français XVII^{e}). ML : biLSTM.

	S		**Pr**		**PP**		**log(PP)**	
	corrélation	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**	**corrélation**	**p-value**
n=2	-0,061	0,508	0,146	0,113	0,002	0,985	-0,060	0,513
n=3	-0,075	0,413	0,101	0,273	-0,026	0,779	-0,111	0,228
n=4	-0,048	0,601	-0,045	0,622	-0,011	0,909	-0,078	0,396
n=5	-0,040	0,661	-0,106	0,250	-0,038	0,681	-0,014	0,878
n=6	-0,045	0,622	-0,010	0,917	-0,048	0,605	-0,041	0,660
n=7	-0,106	0,251	0,064	0,489	-0,005	0,955	-0,021	0,823
n=8	-0,086	0,350	-0,057	0,539	0,022	0,815	-0,076	0,410
n=9	-0,017	0,855	-0,033	0,721	-0,007	0,940	0,011	0,908
n=10	0,202	**0,027**	-0,025	0,786	-0,027	0,767	0,012	0,898

TABLE 5 – Corrélations et *p-values* calculées entre les métriques d'estimation et le *CER*.

La table 5 montre les corrélations et les *p-values* calculées entre les métriques d'estimation et les *CER*. Une corrélation n'est toutefois significative que si la *p-value* est inférieure à un seuil, ceci traduisant que la relation de corrélation a peu de chances d'être due au hasard ; il s'agit d'un test de corrélation. Nous choisissons ici le seuil de $0,1$ qui rend compte d'une faible présomption contre l'hypothèse nulle. Les résultats de la table 5 montre des *p-values* presque toutes supérieures à ce seuil, ce qui signifie que s'il y a corrélation, elle n'est pas significative. Ces résultats semblent donc réfuter l'hypothèse initiale selon laquelle les probabilités des modèles de langue auraient pu être agrégées pour se substituer à un *CER* exigeant une vérité de terrain.

4.3 Les modèles de langue sont-ils inadaptés ?

	ML probabilités conditionnelles	**ML LSTM**	**ML biLSTM**
n=2	90	14721	257646757092
n=3	126	1010690	235913940342
n=4	426	318251055	221055920422
n=5	1091	723946838	211044617070
n=6	1978	690749546	204520506752
n=7	2801	669397958	200184237186
n=8	3510	655634987	1161841181775
n=9	3940	647905538	13807745026062
n=10	4205	643364471	14481238375005

TABLE 6 – Moyennes des perplexités des modèles de langue sur le sous-corpus de test.

Les résultats précédents suggèrent que i) soit les modèles de langue sont de mauvaise qualité, ii)

soit le corpus de l'étude présente des spécificités particulières ou iii) soit les deux raisons précitées concourent à cette impasse.

Les modèles de langue ont été appris sur les œuvres de Bossuet, Chapelin, Ellain et Gournay. On peut donc les évaluer en calculant leur perplexité sur le sous-corpus des œuvres de Papin, Pascal, Sales et Viau. La table 6 présente les moyennes des perplexités des modèles de langue de l'étude calculées sur les vérités de terrain. Les modèles de langue LSTM et biLSTM présentent des perplexités aberrantes (ils sont non adaptés à la tâche) alors que seuls les modèles de langue à probabilités conditionnelles, pour $n \in [2; 4]$, présentent une meilleure qualité. Nous concluons donc que la mauvaise qualité des modèles de langue explique la non corrélation entre les estimateurs et le *CER*.

5 Conclusion

La mauvaise qualité des modèles de langue ne permet pas de valider ou réfuter notre hypothèse, selon laquelle agréger les probabilités des modèles de langue permettrait d'estimer la qualité d'une sortie d'OCR. Pour en faire l'expérience, il s'agirait de renouveler ces tests avec un ensemble plus vaste de transcriptions d'imprimés du XVII^{e} siècle. Nous cherchions à proposer une alternative au manque de vérités de terrain mais nous constatons qu'un ensemble de 108 pages (16 315 mots) est insuffisant. Si cela ne contredit pas l'éventuelle pertinence des estimateurs envisagés, un ensemble conséquent de données textuelles en français du XVII^{e} siècle reste nécessaire au bon apprentissage des modèles langue. Nous rassemblerons donc plus de données textuelles en français du XVII^{e} siècle pour reconduire l'expérience avec des modèles de langue de meilleure qualité.

Les programmes, en Python 3, sont mis à disposition sur : https://github.com/jbtanguy/RECITAL2020.

Remerciements

Ce travail n'aurait pas été possible sans l'aide de Simon Gabay, Gaël Lejeune et Alice Millour.

Références

Berg-Kirkpatrick T. & Klein D. (2014). Improved Typesetting Models for Historical OCR. In K. Toutanova & H. Wu, Éds., *Actes de 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, volume 2, p. 118–123, Baltimore, Maryland, États-Unis : Association for Computational Linguistics. Anthologie ACL : P14-2020.

Bermes E. (2020). *Le numérique en bibliothèque : naissance d'un patrimoine : l'exemple de la Bibliothèque nationale de France (1997-2019)*. Thèse de doctorat, Paris, Ecole nationale des chartes.

Breuel T. M., Ul-Hasan A., Al-Azawi M. A. & Shafait F. (2013). High-performance OCR for printed English and Fraktur using LSTM networks. In *Actes de 12th International Conference on Document Analysis and Recognition*, ICDAR'13, p. 683–687, Washington, DC, États-Unis : IEEE IEEE Computer Society.

Chen S. F., Beeferman D. & Rosenfeld R. (1998). Evaluation metrics for language models. In *Actes de DARPA Broadcast News Transcription and Understanding Workshop*, p. 275–280, Lansdowne, Virginia, États-Unis : Carnegie Mellon University.

Gabay S. (2019). OCRising 17th French prints. https://editiones.hypotheses.org/1958.

Gupta A., Gutierrez-Osuna R., Christy M., Capitanu B., Auvil L., Grumbach L., Furuta R. & Mandell L. (2015). Automatic assessment of OCR quality in historical documents. In *Actes de Twenty-Ninth AAAI Conference on Artificial Intelligence*, p. 1735–1741, Austin, Texas, États-Unis.

Kiessling B. (2019). Kraken-an universal text recognizer for the humanities. In ADHO, Éd., *Actes de Digital Humanities Conference 2019 - DH2019*, Utrecht, Pays-Bas.

Lejeune G. & Abiven K. (2019). Analyse automatique de documents anciens : tirer parti d'un corpus incomplet, hétérogène et bruité. *Information Retrieval, Document and Semantic Web*, **19**(1).

Smith R. (2007). An overview of the Tesseract OCR engine. In *Actes de Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, p. 629–633, Parana, Brésil : IEEE.

Springmann U., Fink F. & Schulz K. U. (2016). Automatic quality evaluation and (semi-) automatic improvement of OCR models for historical printings. arXiv preprint : 1606.05157.

Springmann U. & Lüdeling A. (2016). OCR of historical printings with an application to building diachronic corpora : A case study using the RIDGES herbal corpus. arXiv preprint : 1608.02153.

Springmann U., Najock D., Morgenroth H., Schmid H., Gotscharek A. & Fink F. (2014). OCR of historical printings of Latin texts : problems, prospects, progress. In *Actes de First International Conference on Digital Access to Textual Cultural Heritage (DATeCH'14)*, p. 71–75, New York, NY, États-Unis : Association for Computing Machinery.

Springmann U., Reul C., Dipper S. & Baiter J. (2018). Ground Truth for training OCR engines on historical documents in German Fraktur and Early Modern Latin. arXiv preprint : 1809.05501.

Ul-Hasan A., Bukhari S. S. & Dengel A. (2016). Ocroract : A sequence learning ocr system trained on isolated characters. In *Actes de 12th IAPR Workshop on Document Analysis Systems (DAS)*, p. 174–179, Santorini, Grèce : IEEE.

Vamvakas G., Gatos B., Stamatopoulos N. & Perantonis S. J. (2008). A complete optical character recognition methodology for historical documents. In *Actes de Eighth IAPR International Workshop on Document Analysis Systems*, p. 525–532, Nara, Japon : IEEE.

Wick C., Reul C. & Puppe F. (2018). Comparison of OCR Accuracy on Early Printed Books using the Open Source Engines Calamari and OCRopus. *ACM/IEEE Joint Conference on Digital Libraries 2018 (JCDL 2018)*, **33**(1), 79–96.