Actes de la 26e conférence sur le Traitement Automatique des Langues Naturelles

Emmanuel Morin, Sophie Rosset, Pierre Zweigenbaum

AfIA

Association française
pour l'Intelligence Artificielle

TALN-RECITAL

Conférence sur le Traitement Automatique des Langues Naturelles

PFIA 2019

Table des matières

Emmanuel Morin, Sophie Rosset et Pierre Zweigenbaum (TALN)
Anne-Laure Ligozat et Sahar Ghannay (RECITAL).
Éditorial 7

.
Comités 8

Volume I : Articles longs

Syrielle Montariol et Alexandre Allauzen.
Apprentissage de plongements de mots dynamiques avec régularisation de la dérive 13

Victor Connes et Nicolas Dugué.
Apprentissage de plongements lexicaux par une approche réseaux complexes 27

Ludovic Tanguy, Pauline Brunet et Olivier Ferret.
Comparaison qualitative et extrinsèque d'analyseurs syntaxiques du français : confrontation de modèles distributionnels sur un corpus spécialisé 39

Loïc Vial, Benjamin Lecouteux et Didier Schwab.
Compression de vocabulaire de sens grâce aux relations sémantiques pour la désambiguïsation lexicale 55

Natalia Grabar, Cyril Grouin, Thierry Hamon et Vincent Claveau.
Corpus annoté de cas cliniques en français 71

Antoine Caubrière, Natalia Tomashenko, Yannick Estève, Antoine Laurent et Emmanuel Morin.
Curriculum d'apprentissage : reconnaissance d'entités nommées pour l'extraction de concepts sémantiques 85

Anissa Hamza et Delphine Bernhard.
Détection des ellipses dans des corpus de sous-titres en anglais 99

Tim Van de Cruys.
La génération automatique de poésie en français 113

Marco Dinarelli et Loïc Grobol.
Modèles neuronaux hybrides pour la modélisation de séquences : le meilleur de trois mondes 127

Amalia Todirascu, Marion Cargill et Thomas Francois.
PolylexFLE : une base de données d'expressions polylexicales pour le FLE 143

Volume II : Articles courts

Kate Thompson, Nicholas Asher, Philippe Muller et Jeremy Auguste.
Analyse faiblement supervisée de conversation en actes de dialogue 159

Salima Mdhaffar, Yannick Estève, Nicolas Hernandez, Antoine Laurent et Solen Quiniou.
Apport de l'adaptation automatique des modèles de langage pour la reconnaissance de la parole : évaluation qualitative extrinsèque dans un contexte de traitement de cours magistraux 167

Sonia Badene, Kate Thompson, Jean-Pierre Lorré et Nicholas Asher.
Apprentissage faiblement supervisé de la structure discursive 175

Frédéric Béchet, Cindy Aloui, Delphine Charlet, Géraldine Damnati, Johannes Heinecke, Alexis Nasr et Frédéric Herlédan.
CALOR-QUEST : un corpus d'entraînement et d'évaluation pour la compréhension automatique de textes 185

Iris Eshkol-Taravella, Mariame Maarouf, Marie Skrovec et Flora Badin.
Chunker différents types de discours oraux : défis pour l'apprentissage automatique 195

Yuming Zhai, Gabriel Illouz et Anne Vilnat.

Classification automatique des procédés de traduction 205

Guillaume Wisniewski.
Combien d'exemples de tests sont-ils nécessaires à une évaluation fiable ? Quelques observations sur l'évaluation de l'analyse morpho-syntaxique du français. 215

Tsanta Randriatsitohaina et Thierry Hamon.
De l'extraction des interactions médicament-médicament vers les interactions aliment-médicament à partir de textes biomédicaux : Adaptation de domaine 223

Fiammetta Namer, Lucie Barque, Olivier Bonami, Pauline Haas, Nabil Hathout et Delphine Tribout.
Demonette2 - Une base de données dérivationnelles du français à grande échelle : premiers résultats 233

Elise Bigeard et Natalia Grabar.
Détecter la non-adhérence médicamenteuse dans les forums de discussion avec les méthodes de recherche d'information 245

Rémi Cardon et Natalia Grabar.
Détection automatique de phrases parallèles dans un corpus biomédical comparable technique/simplifié 255

Benoît Sagot.
Développement d'un lexique morphologique et syntaxique de l'ancien français 265

Adrien Bardet, Fethi Bougares et Loïc Barrault.
Étude de l'apprentissage par transfert de systèmes de traduction automatique neuronaux 275

Antoine Perquin, Gwénolé Lecorvé, Damien Lolive et Laurent Amsaleg.
Évaluation objective de plongements pour la synthèse de parole guidée par réseaux de neurones 285

Sara Meftah, Nasredine Semmar, Youssef Tamaazousti, Hassane Essafi et Fatiha Sadat.
Exploration de l'apprentissage par transfert pour l'analyse de textes des réseaux sociaux 293

Syrielle Montariol, Aina Garí Soler et Alexandre Allauzen.
Exploring sentence informativeness 303

Fréjus A. A. Laleye, Antonia Blanié, Antoine Brouquet, Dan Benhamou et Gaël de Chalendar.
Hybridation d'un agent conversationnel avec des plongements lexicaux pour la formation au diagnostic médical 313

Nadia Bebeshina-Clairet et Mathieu Lafourcade.
Inférence des relations sémantiques dans un réseau lexico-sémantique multilingue 323

Jean-Yves Antoine, Marion Crochetet, Céline Arbizu, Emmanuelle Lopez, Samuel Pouplin, Amélie Besnier et Mathieu Thebaud.
Ma copie adore le vélo : analyse des besoins réels en correction orthographique sur un corpus de dictées d'enfants 333

Olga Seminck, Vincent Segonne et Pascal Amsili.
Modèles de langue appliqués aux schémas Winograd français 343

Patricia Chiril, Farah Benamara, Véronique Moriceau, Marlène Coulomb-Gully et Abhishek Kumar.
Multilingual and Multitarget Hate Speech Detection in Tweets 351

Iris Eshkol-Taravella et Hyun Jung Kang.
Observation de l'expérience client dans les restaurants 361

Laurent Kevers, Florian Guéniot, A. Ghjacumina Tognotti et Stella Retali-Medori.
Outiller une langue peu dotée grâce au TALN : l'exemple du corse et de la BDLC 371

Amira Barhoumi, Nathalie Camelin, Chafik Aloulou, Yannick Estève et Lamia Hadrich Belguith.
Plongements lexicaux spécifiques à la langue arabe : application à l'analyse d'opinions 381

Saoussen Mathlouthi Bouzid et Chiraz Ben Othmane Zribi.
Q-learning pour la résolution des anaphores pronominales en langue arabe 391

Tom Bourgeade et Philippe Muller.
Représentation sémantique distributionnelle et alignement de conversations par chat 399

Quentin Gliosca et Pascal Amsili.
Résolution des coréférences neuronale : une approche basée sur les têtes 409

Amir Hazem, Béatrice Daille, Dominique Stutzmann, Jacob Currie et Christine Jacquin.
Réutilisation de textes dans les manuscrits anciens 417

Aleksandra Miletić, Delphine Bernhard, Myriam Bras, Anne-Laure Ligozat et Marianne Vergez-Couret.
Transformation d'annotations en parties du discours et lemmes vers le format Universal Dependencies : étude de cas pour l'alsacien et l'occitan 427

Yoann Dupont.
Un corpus libre, évolutif et versionné en entités nommées du français 437

Filipo Studzinski Perotto, Fadila Taleb, Eric Trupin, Youssouf Saidali, Maryvonne Holzem, Jacques Labiche et Laurent Vercouter.
Une approche hybride pour la segmentation automatique de documents juridiques 447

Volume III : RECITAL

Mathilde Regnault.
Adaptation d'une métagrammaire du français contemporain au français médiéval 459

Mérième Bouhandi.
Apport des termes complexes pour enrichir l'analyse distributionnelle en domaine spécialisé 473

Jessica López Espejel.
Automatic summarization of medical conversations, a review 487

Bruno Oberle.
Détection automatique de chaînes de coréférence pour le français écrit : règles et ressources adaptées au repérage de phénomènes linguistiques spécifiques 499

Ygor Gallina.
Etat de l'art des méthodes d'apprentissage profond pour l'extraction automatique de termes-clés 513

Emmanuelle Kelodjoue.
Extraction d'opinions pour l'analyse multicritère à partir de corpus oraux transcrits : État de l'art 525

Léon-Paul Schaub et Cyndel Vaudapiviz.
Les systèmes de dialogue orientés-but : état de l'art et perspectives d'amélioration 541

Mathilde Veron.
Lifelong learning et systèmes de dialogue : définition et perspectives 563

Manon Scholivet.
Méthodes de représentation de la langue pour l'analyse syntaxique multilingue 577

Dusica Terzic.
Parsing des textes journalistiques en serbe à l'aide du logiciel Talismane 591

Sandra Bellato.
Vers la traduction automatique d'adverbiaux temporels du français en langue des signes française 605

Volume IV : Démonstrations

Didier Schwab, Pauline Trial, Céline Vaschalde, Loïc Vial et Benjamin Lecouteux.
Apporter des connaissances sémantiques à un jeu de pictogrammes destiné à des personnes en situation de handicap : un ensemble de liens entre WordNet et Arasaac, Arasaac-WN 619

Guillaume Dubuisson Duplessis, Sofiane Kerroua, Ludivine Kuznik et Anne-Laure Guénet.
Cameli @ : analyses automatiques d'e-mails pour améliorer la relation client 623

Marine Schmitt, Élise Moreau, Mathieu Constant et Agata Savary.
Démonstrateur en-ligne du projet ANR PARSEME-FR sur les expressions polylexicales 627

Olivier Hamon, Kévin Espasa et Sara Quispe.
SylNews, un agréfilter multilingue 631

Ioan Calapodescu, Caroline Brun, Vassilina Nikoulina et Salah Aït-Mokhtar.
"Sentiment Aware Map" : exploration cartographique de points d'intérêt via l'analyse de sentiments au niveau des aspects 635

Alexandre Arnold, Gérard Dupont, Catherine Kobus, François Lancelot et Pooja Narayan.
Interprétation et visualisation contextuelle de NOTAMs (messages aux navigants aériens) . . . 639

Éditorial

La 26e édition de la conférence TALN et la 21e édition de la session jeunes chercheuses et chercheurs RECITAL se déroulent cette année à Toulouse au sein de la Plateforme française d'intelligence artificielle (PFIA). TALN a une longue tradition de tenue conjointe avec des conférences de domaines proches. Cette pratique a été initiée avec les Journées d'étude sur la parole (JEP) en 2002 à Nancy puis depuis 2008 tous les quatre ans (2008 : Avignon, 2012 : Grenoble, 2016 : Paris). Elle s'est diversifiée avec la Conférence de recherche d'information et applications (CORIA) en 2018 à Rennes. Elle innove cette année avec un hébergement à Toulouse au sein de PFIA. Ces événements sont l'occasion de rencontres enrichissantes pour tous. Cette année, ce ne sont pas moins de huit conférences, sans compter les ateliers associés, aux sessions desquelles les participants à TALN-RECITAL pourront se mêler : APIA (5e Conférence sur les Applications Pratiques de l'Intelligence Artificielle), CAp (21e Conférence sur l'Apprentissage Automatique), IC (30es Journées Francophones Ingénierie des Connaissances), JFPDA (14es Journées Planification, Décision et Apprentissage), JFSMA (27es Journées Francophones sur les Systèmes Multi-Agents), JIAF (13es Journées d'Intelligence Artificielle Fondamentale), RJCIA (17e Rencontre des Jeunes Chercheurs en Intelligence Artificielle), ainsi que CNIA (22e Conférence Nationale en Intelligence Artificielle), qui regroupe les thématiques de l'intelligence artificielle non couvertes par les conférences précédentes.

Les conférences invitées plénières, les sessions de présentations affichées et de démonstrations, les déjeuners et pauses café, les dîners de la conférence sont autant de moments programmés pour que se retrouvent les participants de toutes les conférences. Nous tenons à saluer la qualité de la planification et du suivi du comité scientifique de la plateforme ainsi que le grand travail du comité d'organisation, le tout visant à assurer que l'ensemble des conférences se tienne dans les meilleures conditions et au meilleur coût.

Pour la deuxième année consécutive, les modalités de soumission à TALN se faisaient avec un appel unique et un seul format de soumission en article court pouvant être étendu en article long sur proposition du comité de programme (et demande préalable des auteurs). Nous avons ainsi reçu soixante cinq articles courts et le comité de programme a proposé à dix articles le passage en format long (15 %) et a retenu trente et un articles en format court (48 %). Chaque article a été relu par trois membres du comité de lecture en s'appuyant le cas échant sur des relecteurs additionnels. Le comité de programme s'est appuyé sur ces relectures pour sélectionner lors d'une réunion pleinière les articles composant le programme. C'est un fonctionnement auquel nous sommes profondément attachés pour assurer une diversité dans les thématiques abordées. L'ensemble des évaluations ont été réalisées en double aveugle. Nous remercions les membres des comités de programme et de lecture (à parité femme – homme) pour leur contribution indispensable à ce processus. Le programme de la conférence est complété par quatre démonstrations sélectionnées par le comité de programme. Les titres des sessions donnent une idée des thématiques abordées par la conférence. Ils comprennent des paliers et tâches habituels du TAL (Morphologie et Syntaxe, Syntaxe, Résolution d'anaphores, Multilinguisme), reflètent la place prise par l'apprentissage (Apprentissage par transfert et modèles de langue, Plongements de mots), l'importance fondamentale que continuent à jouer les corpus et bases de données lexicales (Ressources), et l'intérêt du TAL pour des domaines particuliers (Langues spécialisées, Traitement de la langue biomédicale). Comme chaque année, l'ATALA a décerné un prix de thèse dont la récipiendaire présentera son travail en session plénière. La conférence a invité la présentation d'instruments récents du CNRS par leurs coordinatrices : d'une part le pré-GDR TAL (INS2I / informatique), qui adopte une vision inclusive du traitement de la langue (écrite, orale, signée), couvrant les communautés du traitement automatique des langues, du traitement automatique du langage parlé et de la recherche d'information ; d'autre part le GDR LIFT (INSHS / sciences du langage) sur la linguistique informatique, formelle et de terrain.

Cette année, dix-sept articles ont été soumis à RECITAL. Après avoir été chacun évalué par deux membres du comité de programme, quatre articles ont été retenus pour une présentation orale (soit un taux de sélection pour présentation orale de 24 %), et sept autres ont été retenus pour une présentation sous forme de poster (taux de sélection global de 65 %). Nous avons ainsi pu donner l'opportunité à douze jeunes chercheuses et chercheurs, en grande majorité en début de thèse, de présenter leurs travaux à la communauté. Nous remercions le comité de programme (également à parité femme – homme) pour leur minutieux travail de relecture.

Nous souhaitons pour finir au public de ces conférences une semaine riche en découvertes scientifiques et en rencontres de nouveaux collègues, dans une ambiance assurément chaude pour toute la semaine.

Emmanuel Morin, Sophie Rosset et Pierre Zweigenbaum (TALN)
Anne-Laure Ligozat et Sahar Ghannay (RECITAL)

Comités

Présidents de TALN

— Emmanuel Morin (LS2N, Université de Nantes)
— Sophie Rosset (LIMSI, CNRS, Université Paris-Saclay)
— Pierre Zweigenbaum (LIMSI, CNRS, Université Paris-Saclay)

Membres du CP de TALN

— Delphine Bernard (LiLPa, Université de Strasbourg)
— Chloé Braud (LORIA, CNRS)
— Nathalie Camelin (LIUM, Le Mans Université)
— Peggy Cellier (IRISA, INSA Rennes)
— Benoît Crabbé (LLF, Université Paris Diderot)
— Iris Eshkol-Taravella (MoDyCo, Université Paris Nanterre)
— Cécile Fabre (CLLE-ERSS, Université Toulouse - Jean Jaurès)
— Núria Gala (LPL, Aix Marseille Université)
— Thierry Hamon (LIMSI, Université Paris Nord)
— Philippe Langlais (RALI/DIRO, Université de Montréal)
— Gwénolé Lecorvé (IRISA, Université de Rennes 1)
— Aurélie Névéol (LIMSI, CNRS, Université Paris-Saclay)
— Damien Nouvel (ERTIM, INaLCO)
— Didier Schwab (LIG, Université Grenoble Alpes)
— Xavier Tannier (LIMICS, Université Pierre et Marie Curie)

Comité de lecture de TALN

— Gilles Adda (LIMSI, CNRS, Université Paris-Saclay)
— Salah Ait-Mokhtar (Naver Labs Europe)
— Alexandre Allauzen (LIMSI, CNRS, Université Paris-Saclay)
— Maxime Amblard (LORIA, Université de Lorraine)
— Jean-Yves Antoine (LIFAT, Université de Tours)
— Loïc Barrault (LIUM, Le Mans Université)
— Denis Béchet (LS2N, Université de Nantes)
— Frederic Béchet (LIS, Aix-Marseille Université)
— Patrice Bellot (LIS, Aix-Marseille Université)
— Asma Ben Abacha (Lister Hill Center, National Library of Medicine)
— Laurent Besacier (LIG, Université Grenoble Alpes)
— Yves Bestgen (ILC, Université catholique de Louvain)
— Philippe Blache (LPL, CNRS, Aix-Marseille Université)
— Fethi Bougares (LIUM, Le Mans Université)
— Thierry Charnois (LIPN, Université Paris 13)
— Vincent Claveau (IRISA, CNRS)
— Chloé Clavel (LTCI, Télécom ParisTech)
— Kevin Bretonnel Cohen (University of Colorado School of Medicine)
— Béatrice Daille (LS2N, Université de Nantes)
— Géraldine Damnati (Orange Labs)
— Gaël Dias (GREYC, Normandie Université)
— Marco Dinarelli (LIG, CNRS)
— Patrick Drouin (OLST, Université de Montréal)
— Dominique Estival (MARCS, Western Sydney University)
— Yannick Estève (LIUM, Le Mans Université)
— Olivier Ferret (CEA LIST)
— Karën Fort (STIH, Sorbonne Université)
— Thomas Francois (CENTAL, Université catholique de Louvain)
— Éric Gaussier (LIG, Université Grenoble Alpes)
— Jérôme Goulian (LIG, Université Grenoble Alpes)

— Natalia Grabar (STL, CNRS)
— Cyril Grouin (LIMSI, CNRS, Université Paris-Saclay)
— Olivier Hamon (Syllabs)
— Nabil Hathout (CLLE-ERSS, CNRS)
— Amir Hazem (LS2N, Université de Nantes)
— Nicolas Hernandez (LS2N, Université de Nantes)
— Stéphane Huet (LIA, Université d'Avignon et des Pays de Vaucluse)
— Christine Jacquin (LS2N, Université de Nantes)
— Sylvain Kahane (Modyco, Université Paris Nanterre)
— Olivier Kraif (LIDILEM, Université Grenoble Alpes)
— Mathieu Lafourcade (LIRMM, Université de Montpellier)
— David Langlois (LORIA, Université de Lorraine)
— Eric Laporte (LIGM, Université Paris-Est Marne-la-Vallée)
— Thomas Lavergne (LIMSI, Université Paris Sud, Université Paris-Saclay)
— Joseph Le Roux (LIPN, Université Paris 13)
— Benjamin Lecouteux (LIG, Université Grenoble Alpes)
— Yves Lepage (Waseda University)
— Denis Maurel (LIFAT, Université de Tours)
— Richard Moot (LIRMM, CNRS)
— Véronique Moriceau (IRIT, Université Paul Sabatier)
— Philippe Muller (IRIT, Université Paul Sabatier)
— Alexis Nasr (LIS, Aix Marseille Université)
— Adeline Nazarenko (LIPN, Université Paris 13)
— Luka Nerima (Université de Genève)
— Jian-Yun Nie (RALI/DIRO, Université de Montréal)
— Yannick Parmentier (LORIA, Université de Lorraine)
— Sebastian Peña Saldarriaga (Dictanova)
— Thierry Poibeau (Lattice, CNRS)
— Alain Polguère (ATILF, Université de Lorraine)
— Jean-Philippe Prost (LIRMM, Université de Montpellier)
— Solen Quiniou (LS2N, Université de Nantes)
— Christian Raymond (IRISA, INSA Rennes)
— Christian Retoré (LIRMM, Université de Montpellier)
— Djamé Seddah (ALMAnaCH, Paris Sorbonne Université)
— Gilles Serasset (LIG, Université Grenoble Alpes)
— Michel Simard (NRC, Canada)
— Kamel Smaili (LORIA, Université de Lorraine)
— Pascale Sébillot (IRISA, INSA Rennes)
— Ludovic Tanguy (CLLE-ERSS, Université Toulouse - Jean Jaurès)
— Juan-Manuel Torres-Moreno (LIA, Université d'Avignon et des Pays de Vaucluse)
— Guillaume Wisniewski (LIMSI, Université Paris-Sud, Université Paris-Saclay)
— François Yvon (LIMSI, CNRS, Université Paris-Saclay)

Relecteurs additionnels de TALN

— Jingshu Liu (Dictanova)
— Emile Chapuis (LTCI, Télécom ParisTech)
— Caroline Langlet (LTCI, Paris Sorbonne Université)
— Joseph Lark (Dictanova)
— Alexandre Garcia (LTCI, Télécom ParisTech)

Présidentes de RECITAL

— Anne-Laure Ligozat (LIMSI, CNRS, Université Paris-Saclay)
— Sahar Ghannay (LIMSI, CNRS, Université Paris-Saclay)

Membres du CP de RECITAL

— Jean-Yves Antoine (LIFAT, Université de Tours)

— Ismail Badache (ESPE / LIS, Aix-Marseille Université)
— Amira Barhoumi (LIUM, Université du Maine - MIRACL Sfax)
— Rachel Bawden (University of Edinburgh)
— Aurélien Bossard (LIASD, Université Paris 8)
— Chloé Braud (LORIA, CNRS)
— Nathalie Camelin (LIUM, Université du Maine)
— Rémi Cardon (STL, Lille)
— Peggy Cellier (IRISA, INSA Rennes)
— Antoine Doucet (L3i, Université de la Rochelle)
— Maha Elbayad, LIG/ Inria
— Arnaud Ferré (LIMSI-CNRS/MaIAGE-INRA, Université Paris-Saclay)
— Amel Fraisse (Gériico, Lille)
— Thomas François (CENTAL, Université catholique de Louvain)
— Nicolas Hernandez (LS2N, Université de Nantes)
— Yann Mathet (Greyc, Université de Caen)
— Alice Millour (STIH, Université Paris-Sorbonne)
— Anne-Lyse Minard (LLL, Orléans)
— Jose Moreno (IRIT, UPS)
— Tsanta Randriatsitohaina (LIMSI, Université Paris-Sud, Université Paris-Saclay)
— Loïc Vial (LIG, Université Grenoble Alpes)

Volume I : Articles longs

Apprentissage de plongements de mots dynamiques avec régularisation de la dérive

Syrielle Montariol[1, 2] Alexandre Allauzen[1]
(1) LIMSI, CNRS, Univ. Paris-Sud, Univ. Paris-Saclay, F-91405 Orsay, France
(2) Société Générale, 17 Cours Valmy 92043 Puteaux, France
syrielle.montariol@limsi.fr, alexandre.allauzen@limsi.fr

RÉSUMÉ

L'usage, le sens et la connotation des mots peuvent changer au cours du temps. Les plongements lexicaux diachroniques permettent de modéliser ces changements de manière non supervisée. Dans cet article nous étudions l'impact de plusieurs fonctions de coût sur l'apprentissage de plongements dynamiques, en comparant les comportements de variantes du modèle *Dynamic Bernoulli Embeddings*. Les plongements dynamiques sont estimés sur deux corpus couvrant les mêmes deux décennies, le *New York Times Annotated Corpus* en anglais et une sélection d'articles du journal *Le Monde* en français, ce qui nous permet de mettre en place un processus d'analyse bilingue de l'évolution de l'usage des mots.

ABSTRACT

Learning dynamic word embeddings with drift regularisation

Word usage, meaning and connotation change throughout time. Diachronic word embeddings are used to grasp these changes in an unsupervised way. In this paper, we use variants of the *Dynamic Bernoulli Embeddings* model to learn dynamic word embeddings, in order to identify notable properties of the model. The comparison is made on the *New York Times Annotated Corpus* in English and a set of articles from the French newspaper *Le Monde* covering the same period. This allows us to define a pipeline to analyse the evolution of words use across two languages.

MOTS-CLÉS : Diachronie, Plongements lexicaux, analyse bilingue.

KEYWORDS: Diachrony, word embeddings, cross-lingual analysis.

1 Introduction

Les langues peuvent être considérées comme des systèmes dynamiques : l'usage des mots évolue au cours du temps, reflétant les nombreux aspects des évolutions de la société, qu'ils soient culturels, technologiques ou dûs à d'autres facteurs (Aitchison, 2001).

La diachronie désigne l'étude de ces variations temporelles d'usage et de sens au sein d'une langue. Ici, nous étudions un corpus journalistique d'une plage temporelle de deux décennies : l'usage des mots évolue suite à des événements ayant un retentissement médiatique. Par exemple, l'usage du mot "Katrina" a connu un important changement au cours de ces deux décennies. Si par le passé, il fut exclusivement utilisé comme un prénom féminin, comme *Justine* et *Sonja*, dès 1999, son sens se rapproche de celui d'*ouragan*, avec l'arrivée du premier orage tropical éponyme. Puis à partir de

2005 où le cyclone Katrina eut lieu, ce qui fut un prénom féminin partage désormais le même champs lexical que les mots "désastre", "dévastation" et "inondation".

Détecter et comprendre ces changements avec le concours de méthodes d'apprentissage automatique est utile à la recherche linguistique, mais aussi à de nombreuses tâches de traitement automatique des langues. Ajouter une notion temporelle aux représentations de mots permet d'étudier des corpus qui s'étendent sur des plages temporelles longues avec une plus grande acuité. Le problème se pose particulièrement aujourd'hui, alors qu'un nombre croissant de documents historiques sont numérisés et rendus accessibles ; leur analyse conjointe à celle de corpus contemporains, pour des tâches allant de la classification de documents à la recherche d'information, nécessite de prendre en compte la diachronie.

Suivant les travaux de Bengio *et al.* (2003) puis Mikolov *et al.* (2013), de nombreuses méthodes de représentations vectorielles de mots ont été mises au point depuis deux décennies. Elles permettent de représenter les mots par des vecteurs continus de faible dimension : les plongements lexicaux ou *word embeddings*. Néanmoins, ces plongements lexicaux reposent sur l'hypothèse que le sens d'un mot est inchangé sur l'ensemble du corpus. Cette hypothèse d'une représentation statique peut s'avérer limitée. Ainsi en supposant qu'un changement dans le contexte usuel d'un mot reflète un changement dans la signification de ce mot, il est possible d'entraîner des plongements de mots diachroniques : qui évoluent au cours du temps en suivant les changements d'usage des mots.

Récemment Rudolph & Blei (2018) ont proposé un tel modèle, nommé *Dynamic Bernoulli Embedding* (DBE). Il apprend des représentations de mots qui évoluent au cours du temps selon les strate temporelles d'un corpus, en caractérisant la dérive de ces représentation d'une strate à l'autre au moyen d'un processus aléatoire gaussien. Des choix de modélisation différents ont été effectués par d'autres auteurs dans la littérature. Les approches de Han *et al.* (2018) et Hamilton *et al.* (2016) impliquent d'apprendre des plongements lexicaux pour chaque strate temporelle sans les relier chronologiquement ; tandis que Kim *et al.* (2014) apprend les plongements diachroniques de façon incrémentale, mais sans contrôler la dérive de ces plongements.

Dans cet article nous prenons pour base le modèle DBE, qui présente un bon compromis entre simplicité et modulabilité, pour questionner l'importance de ces différents choix de modélisation. Dans ce but, nous analysons le comportement des plongements de mots appris à partir de ce modèle (décrit à la section 3) sur deux tailles de strates temporelles – mensuelle et annuelle – et l'appliquons (dans la section 4) à des corpus dans deux langues différentes : français et anglais. Les données en anglais proviennent du *New York Times Annotated Corpus* [1] (Sandhaus, 2008), qui s'étend de 1987 à 2006 ; le corpus en français est constitué d'articles du journal *LeMonde* collectés de façon à couvrir la même période. L'étude de ces deux corpus nous permet, dans un deuxième temps, d'étudier de façon conjointe l'évolution d'un mot à travers les deux langages.

2 État de l'art

Les premières méthodes automatiques d'étude de la diachronie se basent sur la détection de changements dans les co-occurrences des mots, puis sur des approches basées sur la similarité distributionnelle (Gulordava & Baroni, 2011) en construisant des mesures d'information mutuelle à partir de matriças de co-occurrences.

1. https ://catalog.ldc.upenn.edu/LDC2008T19

L'usage de méthodes d'apprentissage automatique basées sur les plongements lexicaux est récent et a connu une forte hausse d'intérêt depuis deux ans, avec la publication consécutive de trois articles dédiés à l'état de l'art de ce domaine (Kutuzov *et al.*, 2018; Tahmasebi *et al.*, 2018; Tang, 2018).

Dans un des premiers articles employant ce type de méthode (Kim *et al.*, 2014), les auteurs estiment des plongements lexicaux pour la première strate temporelle t_0 puis mettent à jour ces plongements pour les strates temporelles suivantes, considérant les plongements au temps $t-1$ comme initialisation pour la strate t. D'autres travaux ont ensuite vu le jour, reposant sur l'apprentissage de façon indépendante des plongements lexicaux pour chaque strate temporelle. Néanmoins, les plongements ainsi obtenus ne sont pas directement comparables car appartiennent à des espaces vectoriels différents. Deux approches sont alors envisageables : d'une part, déterminer la meilleure transformation linéaire afin d'aligner les espaces de représentation à travers les périodes (Hamilton *et al.*, 2016; Dubossarsky *et al.*, 2017; Szymanski, 2017; Kulkarni *et al.*, 2015) ; d'autre part, calculer la similarité cosinus entre chaque paire de mot à l'intérieur d'une strate temporelle, les similarités étant alors comparables d'une strate sur l'autre sans nécessiter d'alignement (Kim *et al.*, 2014).

Les méthodes dites dynamiques constituent un second type d'approche. Le corpus d'étude est toujours divisé en strates temporelles, mais cette fois les plongements lexicaux diachroniques sont appris de façon conjointe sur l'ensemble des strates. Ils sont ainsi placés dans un même espace de représentation dès l'apprentissage. Pour cela, Bamler & Mandt (2017) utilisent des modèles bayésiens d'apprentissage de plongements lexicaux : les vecteurs sont liées à travers les périodes à l'aide d'un processus de diffusion temporel qui contrôle leur évolution. Poursuivant le même objectif, différentes méthodes ont été proposées (Yao *et al.*, 2018; Rudolph & Blei, 2018; Han *et al.*, 2018) afin de mettre en évidence de façon jointe l'évolution continue du sens des mots. Ces méthodes permettent de s'affranchir de la limite de volume de données par strate temporelle lors de l'apprentissage.

La majorité de ces modèles sont évalués sur des corpus en anglais. À notre connaissance, bien que plusieurs auteurs ont expérimenté sur d'autres langues que l'anglais (Hamilton *et al.*, 2016; Eger & Mehler, 2016), aucun travaux n'a tenté de comparer l'évolution de mots à travers plusieurs langues à l'aide de méthodes de plongements diachroniques.

3 Modèles de plongements de mots dynamiques

Nous partons du modèle *Dynamic Bernoulli Embeddings* (DBE) de Rudolph & Blei (2018). Il se base sur les plongements de mots de la famille exponentielle (Rudolph *et al.*, 2016), qui sont une généralisation probabiliste du modèle *Continuous Bag-of-Words* (CBOW) de Mikolov *et al.* (2013). Nous en fournissons une brève description avant de présenter les différentes variantes mises en place.

3.1 Le modèle DBE

L'objectif de ce modèle est de prédire un mot à partir de son contexte. Afin de désigner un mot v parmi un vocabulaire de taille V, on considère un vecteur de variables aléatoires binaires $\mathbf{x}_v \in \{0,1\}^V$, où seule la composante associée au mot v vaut 1. Le mot v à la position i dans le corpus est donc représenté par le vecteur binaire $\mathbf{x}_{iv}$ et son contexte $\boldsymbol{c}_i$ est constitué des C mots avant et des C mots après (C étant la taille de la fenêtre). Ainsi, $\mathbf{x}_{\boldsymbol{c}_i}$ regroupe l'ensemble des points constituant le contexte du mot i. Le modèle DBE prédit le vecteur binaire du mot $\mathbf{x}_{iv}$ à partir de son vecteur de contexte $\mathbf{x}_{\boldsymbol{c}_i}$

selon la loi de Bernoulli suivante : $\mathbf{x}_{iv}|\mathbf{x}_{\boldsymbol{c}_i} \sim Bern(p_{iv})$. Le paramètre de la loi de Bernoulli p_{iv} est calculé à partir des plongements lexicaux $\boldsymbol{\rho}_v$ du mot à prédire et $\boldsymbol{\alpha}_{v'}$ des mots du contexte :

$$p_{iv} = \sigma\Big(\boldsymbol{\rho}_v^T(\sum_{j\in\boldsymbol{c}_i}\sum_{v'\in V}\boldsymbol{\alpha}_{v'}\mathbf{x}_{jv'})\Big). \tag{1}$$

Ainsi la somme sur v' sélectionne les plongements $\boldsymbol{\alpha}_{v'}$ des mots du contexte, qui sont ensuite additionnés (somme sur j) afin de créer un vecteur représentant le contexte $\boldsymbol{c}_i$. Le paramètre de Bernoulli résulte de l'application de la fonction sigmoïde σ au produit scalaire de ce vecteur avec le plongement $\boldsymbol{\rho}_v$ du mot à prédire.

Vers un modèle dynamique : Pour rendre ce modèle dynamique, considérons un corpus composé de T strates temporelles indicées par t. Dans chaque strate, chaque mot v a deux types de représentions : celle en tant que mot de contexte $\boldsymbol{\alpha}_v$, et celle en tant que mot central $\boldsymbol{\rho}_v$. Le vecteur $\boldsymbol{\alpha}_v$ est considéré comme invariant : il est commun à toutes les strates temporelles. Seuls les plongements $\boldsymbol{\rho}_v$ évoluent au cours du temps selon la marche aléatoire gaussienne suivante :

$$\boldsymbol{\rho}_v^{(0)} \sim \mathcal{N}(0, \lambda_0^{-1}I), \text{ puis pour } \forall t \geq 1,\ \boldsymbol{\rho}_v^{(t)} \sim \mathcal{N}(\boldsymbol{\rho}_v^{(t-1)}, \lambda^{-1}I). \tag{2}$$

Le paramètre λ, nommé *dérive*, est le même pour l'ensemble des strates et contrôle l'évolution du vecteur $\boldsymbol{\rho}_v$ d'une strate temporelle sur l'autre.

Apprentissage : L'apprentissage de ce modèle, plus précisément décrit par Rudolph & Blei (2018), s'appuie sur une variante de la stratégie du *negative sampling* (Mikolov *et al.*, 2013). L'objectif est d'optimiser la fonction suivante :

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = \mathcal{L}_{pos}(\boldsymbol{\rho}, \boldsymbol{\alpha}) + \mathcal{L}_{neg}(\boldsymbol{\rho}, \boldsymbol{\alpha}) + \mathcal{L}_{prior}(\boldsymbol{\rho}, \boldsymbol{\alpha}) \tag{3}$$

Le premier terme $\mathcal{L}_{pos}$ représente la log-probabilité associée aux exemples positifs, tandis que le second ($\mathcal{L}_{neg}$) correspond à celle associée à des exemples négatifs tirés aléatoirement. Le troisième terme agit comme un terme de régularisation sur $\boldsymbol{\alpha}$ et sur la dynamique des plongements $\boldsymbol{\rho}$, et consiste à pénaliser le vecteur $\boldsymbol{\rho}_v^{(t)}$ lorsqu'il s'éloigne trop fortement du vecteur $\boldsymbol{\rho}_v^{(t-1)}$, de la manière suivante :

$$\mathcal{L}_{prior}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = -\frac{\lambda_0}{2}\sum_v \|\boldsymbol{\alpha}_v\|^2 - \frac{\lambda_0}{2}\sum_v \|\boldsymbol{\rho}_v^{(0)}\|^2 - \frac{\lambda}{2}\sum_{v,t}\|\boldsymbol{\rho}_v^{(t)} - \boldsymbol{\rho}_v^{(t-1)}\|^2. \tag{4}$$

3.2 Variantes de régularisation

La première variante du modèle se rapproche du principe d'apprentissage incrémental proposé par Kim *et al.* (2014). Elle consiste à supprimer la régularisation sur la dérive des plongements de mots. Dans ce cas, la fonction de coût ne prend en compte que les deux premiers termes de la log-priore ainsi que $\mathcal{L}_{pos}$ et $\mathcal{L}_{neg}$. Par la suite, nous intitulons cette variante DBE-I (Incrémental).

La seconde variante consiste à abolir l'obligation de chronologie dans les vecteurs temporels successifs. En remplaçant la troisième composante de $\mathcal{L}_{prior}$ par $\sum_{v,t}\|\boldsymbol{\rho}_v^{(t)} - \boldsymbol{\rho}_v^{(0)}\|^2$, on force le vecteur $\boldsymbol{\rho}_v^{(t)}$ à rester proche du plongement d'origine $\boldsymbol{\rho}_v^{(0)}$. Ce principe est similaire à celui de Han *et al.* (2018), où les plongements de mots diachroniques sont appris de façon indépendante sur chaque strate temporelle. Cette variante est désignée par DBE-NC (Non Chronologique).

Une autre version de cette dernière fonction est mise en place, afin de prendre en compte l'éloignement temporel. Dans ce but, la troisième composante de la log-priore $\sum_{v,t} \|\boldsymbol{\rho}_v^{(t)} - \boldsymbol{\rho}_v^{(0)}\|^2$ est multipliée par un facteur temporel : le coefficient devient $-\frac{\lambda}{2} * t$ et permet de contrôler l'éloignement à la priore sans ajouter de dépendance entre les strates temporelles successives. Cette dernière version est nommée DBE-SC (Semi Chronologique).

4 Expérimentation

Nous expérimentons à partir de notre propre implémentation du modèle DBE en PYTORCH. Dans un premier temps, nous analysons de façon quantitative le comportement des différentes variantes du modèle DBE définies précédemment. Puis nous observons de plus près la dérive des mots ; en particulier, nous mettons en place un processus pour comparer les évolutions des mots dans deux langues de façon conjointe.

4.1 Données et hyper-paramètres

Les données du *New York Times Annotated Corpus* sont composées de 1 855 000 articles s'étalant sur une période d'environs 20 ans, du 1^{er} janvier 1987 au 19 juin 2007. Le journal *Le Monde* est un des quotidiens les plus lus en France ; nous en collectons des articles entre le 1^{er} janvier 1987 et le 31 décembre 2006. Ces deux corpus sont divisés en $T = 20$ strates temporelles annuelles [2] et $T = 240$ strates temporelles mensuelles.

Pour construire le vocabulaire, nous sélectionnons pour les deux langues $V = 40\,000$ mots selon leurs fréquences après avoir retiré les mots-outils. De même que Mikolov *et al.* (2013), nous sous-échantillonnons les mots fréquents en retirant chaque mot i avec une probabilité $p = 1 - \sqrt{\frac{10^{-5}}{\text{fréquence}(i)}}$. Dans le corpus *LeMonde*, le nombre moyen de mots par strate temporelle est d'environ 3.5 millions pour les strates annuelles et 300k pour les strates mensuelles. Dans le corpus *NYT*, ce nombre est d'environ 9 millions pour les strates annuelles et 750k pour les strates mensuelles. Le corpus est ensuite divisé en échantillons d'apprentissage, de validation et de test. Ces derniers comprennent chacune 10 % des données tirées aléatoirement. Les embeddings sont entraînés avec 1000 mini-batchs par strates temporelles pour l'analyse annuelle et 100 mini-batchs pour l'analyse mensuelle.

Les hyper-paramètres sont sélectionnés à partir de l'étude de la log-probabilité sur les exemples positifs $\mathcal{L}_{pos}$ calculée sur l'échantillon de validation de chaque corpus, à partir du modèle DBE classique. Afin de permettre la comparaison, les valeurs de $\mathcal{L}_{pos}$ sont mises à l'échelle selon la règle suivante :

$$\text{Échelle} = \frac{\text{Nb de mots dans l'échantillon de validation}}{\text{Nb de mots dans chaque mini-batch}}.$$

Dans un premier temps, le modèle est entraîné sur l'ensemble du corpus sans composante temporelle (modèle statique). Ainsi, les plongements lexicaux $\boldsymbol{\rho}$ et $\boldsymbol{\alpha}$ peuvent servir par la suite d'initialisation pour les modèles temporels. Suite à l'évaluation sur l'échantillon de validation, la fenêtre de contexte choisie est $C = 4$ [3]. La dimension des plongements lexicaux est de 100 et le nombre d'exemples

2. La dernière année du corpus *NYT* étant incomplète, elle n'est pas prise en compte dans l'analyse.
3. Deux mots précédant et deux mots suivant le mot central.

négatifs tirés pour chaque exemple positif est fixé à 10. Pour finir, la dérive $\lambda = 1$ est celle qui offre les meilleurs résultats. La dérive initiale λ_0 est fixée, comme le font Rudolph & Blei (2018), à $\frac{\lambda}{1000}$.

4.2 Évaluation quantitative

Dans un premier temps, nous analysons l'effet des différentes variantes de la fonction de coût sur les performances du modèle mesurées en terme de log-vraisemblance et sur la distribution des dérives des mots.

4.2.1 Évolution de la log-vraisemblance

Dans cette partie, nous calculons la log-probabilité du modèle DBE sur les exemples positifs $\mathcal{L}_{pos}$ sur les données de test de chaque corpus, *NYT* et *LeMonde*, pour les deux tailles de strates temporelles (Figure 1). Nous appliquons le terme d'échelle décrit dans la partie 4.1.

Dans un premier temps, les plongements lexicaux sont appris sur l'ensemble du corpus de façon statique. Puis ils sont utilisés sur chaque strate annuelle du corpus pour calculer $\mathcal{L}_{pos}$. La courbe obtenue est plus basse que la courbe associée au modèle dynamique appris en initialisant les vecteurs à partir de ce modèle statique. On constate logiquement que l'apprentissage dynamique permet aux plongements d'être adaptés à chaque strate temporelle, et donc plus efficace pour prédire les données de test.

À l'inverse, pour les deux corpus, le modèle dynamique sans initialisation a la performance la plus faible. Cette tendance est confirmée par la log-probabilité moyenne sur l'ensemble des strates temporelles (Table 1). Une explication se trouve peut être dans le faible volume de données sur chaque strate. Quand à la performance du modèle statique, elle dépend de l'homogénéité temporelle du corpus étudié ; nos deux corpus couvrent une plage de temps relativement faible, justifiant la performance du modèle statique par rapport à celle du modèle dynamique sans initialisation.

Comme le montre le tableau 1, les variantes du modèle définies par les différentes fonctions de coût ont des performances très proches ; l'ajout du coefficient de dérive croissant (DBE-SC) au modèle non chronologique (DBE-NC) permet d'augmenter légèrement sa performance, mais dans l'ensemble, c'est le modèle sans régularisation sur la dérive (DBE-I) qui obtient le score le plus élevé quelle que soit la taille de la strate temporelle.

L'étude de la log-probabilité ne reflète qu'une vision globale des performances du modèle ; afin de mieux comprendre son comportement, nous observons ensuite la distribution des dérives pour chaque variation du modèle.

4.2.2 Caractérisation de la dérive des plongmements

Dans le but d'analyser plus en détail le comportement du modèle et l'effet des variations de la fonction de coût, nous représentons les histogrammes superposés des dérives successives observées sur le corpus *LeMonde* (Figure 2). Les histogrammes pour le corpus *NYT* présentent des tendances similaires, de même que le cas des strates temporelles mensuelles. La dérive de chaque mot est calculée à partir de la distance euclidienne entre le plongement du mot au début du corpus $\boldsymbol{\rho}_v^{(t_0)}$ et ses

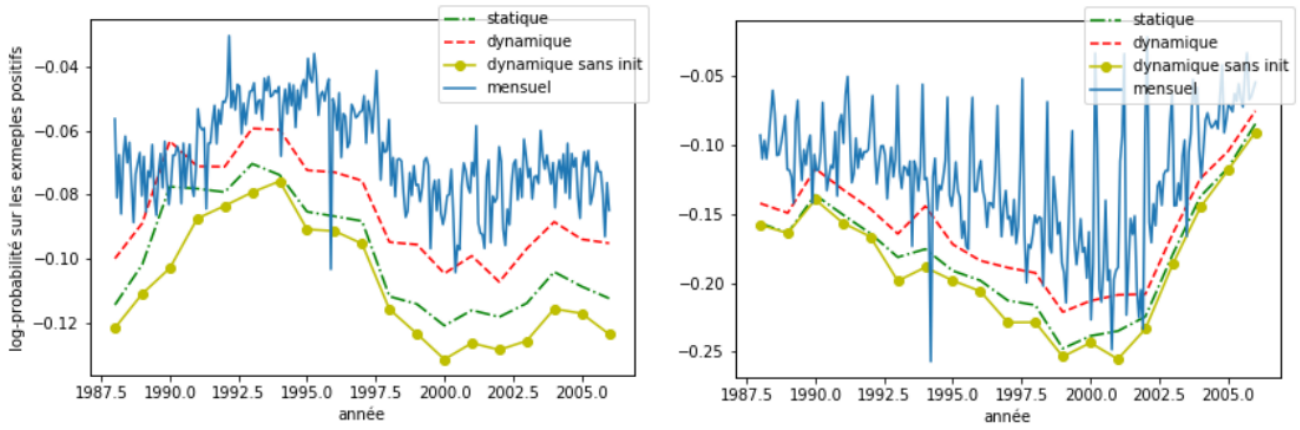

FIGURE 1 – Log-probabilités sur les exemples positifs sur l'échantillon de test du corpus *NYT* (à gauche) et *LeMonde* (à droite), à partir des modèles statique, dynamiques, et dynamiques sans pré-entraînement pour des strates annuelles, et dynamique pour des strates mensuelles.

	NYT		**Le Monde**	
	annuel	mensuel	annuel	mensuel
Statique	-0.09875	-0.06771	-0.1794	-0.1259
DBE	-0.08476	**-0.06720**	-0.1606	-0.1231
DBE sans initialisation	-0.10774	-0.07305	-0.1873	-0.1498
DBE-I	**-0.08448**	-0.06752	**-0.1593**	**-0.1227**
DBE-NC	-0.08517	-0.06817	-0.1607	-0.1236
DBE-SC	-0.08455	-0.06752	-0.1598	-0.1228

TABLE 1 – Log-probabilités moyennes sur l'ensemble des strates temporelles, sur l'échantillon de test des deux corpus, pour les différentes variantes d'apprentissage et de fonction de coût du modèle DBE.

plongements successifs $\boldsymbol{\rho}_v^{(t)}$ à chaque nouvelle strate temporelle t. Sur les histogramme, les couleurs plus claires correspondent aux distributions des dérives aux strates récentes : ainsi, la courbe la plus claire représente la distribution des dérives de mots calculées entre $t_0 = 1987$ et $t = 2006$ tandis que la plus sombre représente la distribution des dérives entre $t_0 = 1987$ et $t = 1988$.

Une première propriété intéressante est le caractère dirigé des dérives. Comme le montre le premier histogramme de la figure 2, les valeurs des dérives augmentent à travers le temps pour le modèle DBE classique. Cela signifie que le modèle capture principalement des dérives possédant une tendance, plutôt que de brefs changements de plongements suivis de retours à la normale. Ainsi le terme de régularisation décrit par l'équation 4 réalise bien le compromis attendu, en considérant comme partie de l'objectif la détection des grandes tendances d'évolution du sens des mots et en omettant leurs brèves variations.

Ces brèves variations sont dues à des évènements qui modifient temporairement le contexte dans lequel apparaît un mot sans avoir un impact à long terme sur son sens. Elles sont plutôt capturées par la version DBE-NC du modèle, dont l'histogramme ne présente pas d'évolution dirigée de la dérive en fonction de la distance à t_0, donc ne distingue pas ces "bruits" de la tendance générale d'évolution des mots. Pour finir, malgré l'absence de terme de régularisation sur la dérive, le modèle DBE-I capture naturellement une dérive relativement dirigée dans le temps bien que l'histogramme montre une plus grande sensibilité au bruit.

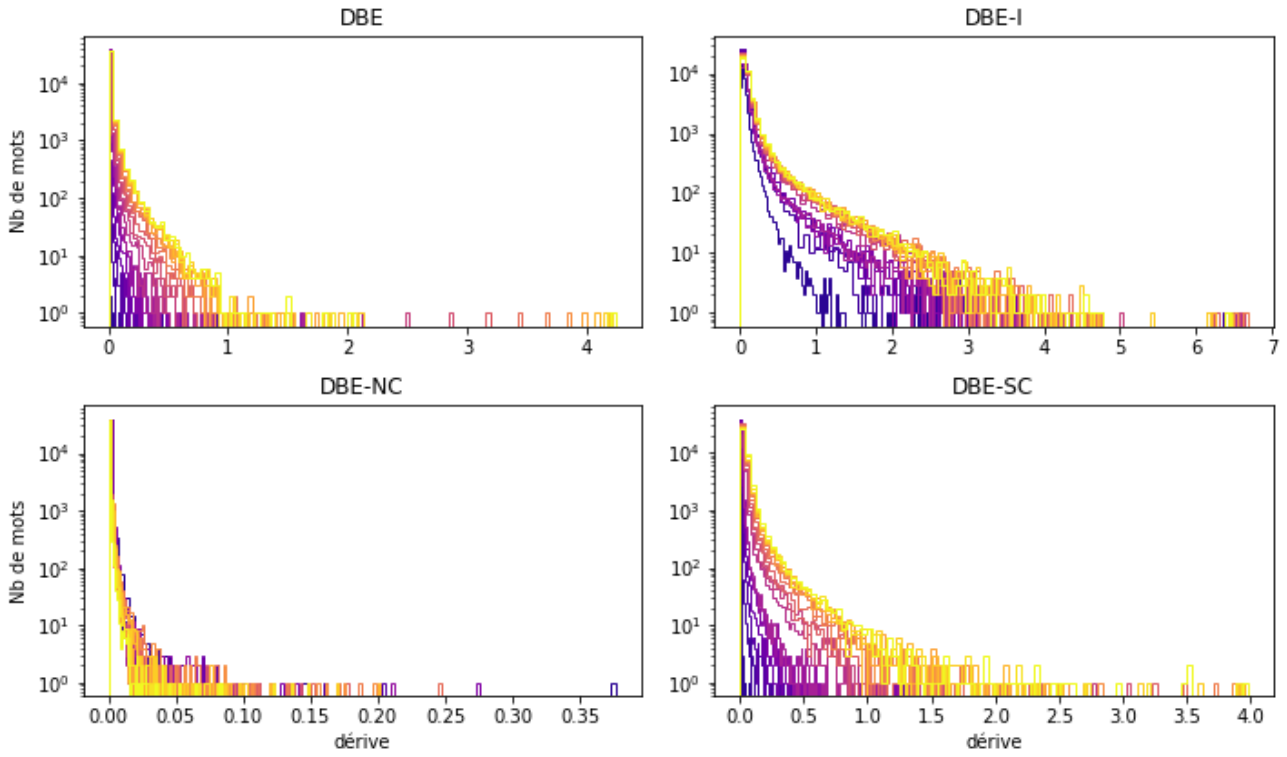

FIGURE 2 – Histogramme des dérives entre les plongements de mots à $t_0 = 1987$ et à chaque strate temporelle annuelle successive du corpus *LeMonde*, pour le modèles DBE et ses trois variantes. Plus la couleur est claire, plus la différence est calculée par rapport à une strate temporelle récente. Les nombres de mots (en ordonnée) sont en échelle logarithmique.

La seconde propriété mise en évidence est la capacité du modèle à distinguer les mots stables des mots dont l'usage évolue. En effet dans un intervalle de deux décennies, la majorité des mots est supposée peu évoluer. Le modèle DBE-NC, en introduisant une régularisation par rapport aux plongements de mots initiaux, permet de forcer le respect cette propriété : une grande part des mots sont presque invariants sur tout le corpus, et seule une sélection de dérives se démarque. Le modèle DBE classique permet aussi, dans une certaine mesure, de garder une faible dérive pour une grande partie des mots ; de même pour le modèle DBE-SC. Seul le modèle DBE-I ne distingue pas naturellement les mots qui dérivent peu.

4.3 Évaluation qualitative

La seconde étape de l'analyse est d'observer directement l'évolution des mots. À notre connaissance, il n'existe pas de corpus annotés permettant une évaluation directe des modèles diachroniques. Il est par contre possible d'observer l'évolution de certains mots choisis, permettant un premier diagnostic sous la forme d'une évaluation qualitative et subjective (Tahmasebi *et al.*, 2018). Ainsi pour chaque variante du modèle, nous observons les mots qui dérivent le plus pour mieux en comprendre le comportement. Puis, afin d'étudier de façon conjointe l'évolution des mots sur les deux corpus, nous mettons en place un processus d'analyse diachronique inter-langues.

NYT	Le Monde					
Annuel	Annuel			Mensuel		
DBE	DBE	DBE-I	DBE-NC	DBE	DBE-I	DBE-NC
google	euros	clearstream	royal	euros	rfa	seznec
skilling	ump	arcelor-mittal	sarkozy	sarkozy	euros	tramway
bloomberg	villepin	raimond	gdf	ump	ségolène	pinochet
email	rfa	shultz	euros	francs	sarkozy	hamas
katrina	sarkozy	ségolène	hezbollah	ségolène	ump	euros
cellphone	al-qaïda	outreau	liban	villepin	villepin	ahmadinejad
darfur	poutine	eads	thaksin	internet	monory	abbas
contras	gorbatchev	zapatero	ump	ue	climatique	révision
blog	katrina	villepin	islam	euro	contras	fibre
euros	internet	zidane	suez	bush	réévaluation	mahmoud

TABLE 2 – Listes des 10 mots ayant la plus grande dérive totale (distance entre la première et la dernière strate temporelle) pour les modèles DBE, DBE-I et DNE-NC sur le corpus *LeMonde* et DBE sur le corpus *NYT*. Les mots ayant un arrière-plan coloré sont communs à plus d'un modèle.

4.3.1 Analyse des fortes dérives

Nous nous concentrons ici sur le corpus *LeMonde*. La période étudiée ne couvrant que deux décennies, on observe principalement des évolutions de contexte liées aux événements ayant un impact médiatique ; les mots subissant de fortes dérives sont en majorité des entités nommées, et sont liés au contexte politique de la période, un thème récurrent dans ce journal.

Nous listons les 10 mots dont l'usage a le plus dérivé au cours des deux décennies selon chaque modèle dans le corpus, pour les deux tailles de strates temporelles. Du corpus *NYT*, nous ne reportons que les 10 mots ayant le plus varié d'après le modèle DBE classique sur des strates temporelles annuelles (Table 2).

Dans le cas du modèle DBE, pour les strates mensuelles et annuelles, les mots qui dérivent le plus sont associés à des concepts ayant subi un changement notable et continu au cours de la période (*euros*, *al-qaïda*, *internet*...). Cette observation est en accord avec la propriété observée précédemment au sujet du caractère dirigé des dérives. À l'inverse, les modèles DBE-I et DBE-NC annuels mettent principalement en valeur des dérives très fortes de mots liées à des événements uniques (*zidane*, *clearstream*, *royal*). C'est particulièrement le cas pour le modèle DBE-NC sur les strates mensuelles , où les dérives rapportées sont sujettes à un bruit important.

Afin de confirmer ces observations, le tableau 3 moyenne pour chaque modèle les rapports entre la dérive moyenne (sur toutes les strates temporelles) et la dérive totale (entre la première et la dernière strate) des 10 et 500 mots qui évoluent le plus. Notons que les valeurs des dérives moyennes normalisées ne sont pas directement comparables, selon que l'on considère les strates annuelles ou mensuelles (car le nombre de strates diffère). La valeur moyenne pour les 10 mots qui dérivent le plus est toujours plus faible que celle sur les 500 mots qui dérivent le plus. Les dérives importantes sont donc les plus dirigées quel que soit le modèle. Le modèle DBE-I présente des valeurs très proches de celles du modèle DBE-SC, montrant que l'absence de régularisation par rapport à la dérive permet, dans une certaine mesure, de conserver une certaine robustesse au bruit, en adéquation avec les observations de la figure 2.

Pour finir, remarquons que parmi les mots qui ont le plus dérivé au cours des deux décennies, certains

	Annuel		Mensuel	
	top 500	**top 10**	**top 500**	**top 10**
DBE	0.00642	0.00785	0.00356	6.280e-05
DBE-I	0.1152	0.0272	0.1149	0.0253
DBE-NC	0.5259	0.1054	0.0767	0.0530
DBE-SC	0.1096	0.0301	0.0715	0.0164

TABLE 3 – Valeurs moyennes pour les 10 mots et les 500 mots dérivant le plus, de leur dérive moyenne normalisée. Les dérives sont calculées pour les 4 variantes du modèle DBE sur le corpus *LeMonde*.

sont communs aux deux corpus (*euros*, *google*, *katrina*). Nous proposons donc par la suite une méthode pour observer l'évolution conjointe de ces mots dans les deux langues.

4.3.2 Analyse conjointe en anglais et français

Dans cette partie, nous étudions l'évolution d'un mots en français dans le corpus *LeMonde* et de sa traduction anglaise dans le corpus *NYT*. Les modèles de plongements de mots étant appris de façon indépendante sur ces deux corpus, les vecteurs ne sont pas directement comparables. Nous effectuons alors un alignement des deux espaces de représentation en utilisant un dictionnaire bilingue comme outil de supervision (Conneau *et al.*, 2018). Dans un premier temps, les plongements de mots des deux langues appris de façon statique sur l'ensemble du corpus sont normalisés ; puis, l'espace de représentation des plongements en français est aligné sur l'espace vectoriel des plongements en anglais. Nous choisissons ce sens car les données du *NYT* sur lesquelles les plongements anglais sont appris ont une volumétrie plus élevée, permettant des plongements lexicaux plus robustes.

Nous utilisons l'outil MUSE [4] pour l'alignement. La supervision est effectuée au moyen d'un dictionnaire bilingue construit à partir des vocabulaires des deux corpus. Nous sélectionnons tous les mots ayant un équivalent dans l'autre langue à partir du dictionnaire fournit par MUSE, puis ajoutons manuellement une sélection de mots spécifiques aux données (principalement des entités nommées). À partir des deux vocabulaires de 40 000 mots, nous obtenons un vocabulaire bilingue de 27 351 mots. Pour finir, les plongements sémantiques alignés $\boldsymbol{\rho}_{\text{align}}$ and $\boldsymbol{\alpha}_{\text{align}}$ sont utilisés pour initialiser les modèles dynamiques entrâinés sur chaque corpus.

Suite à l'apprentissage, pour chaque couple de mot dans le dictionnaire bilingue, nous calculons leurs dérives dans les deux corpus. Puis, nous calculons le cosinus comme une similarité entre les plongements des deux mots à la première et la dernière strate temporelle. Appelons cette valeur la similarité inter-langues. Nous calculons la dérive de cette similarité entre la première et la dernière strate temporelle, en mesurant la distance euclidienne entre ces deux valeurs.

En observant la distribution de ces grandeurs, nous mettons en évidence 4 types de comportement de mot à travers les deux langues :

1. Les mots qui dérivent dans la même direction dans les deux langues ;
2. Les mots qui dérivent dans les deux langages, mais dont le sens diverge (la similarité inter-langues décroît entre le première et la dernière strate temporelle) ;
3. Les mots qui dérivent dans une seule des deux langues, tandis que l'autre reste stable ;
4. Les mots qui sont stables dans les deux langues.

4. https ://github.com/facebookresearch/MUSE

Nous différencions les classes de mots en utilisant la moyenne des dérive des mots de chaque langue ainsi que la moyenne de la dérive de la similarité inter-langues, et reportons leur répartition au sein du vocabulaire bilingue dans le tableau 4. La majorité des mots appartiennent à la catégorie (4) (mots stables dans les deux langues), ce qui confirme la propriété du modèle DBE énoncée à partir de la figure 2 ; tandis que les mots qui évoluent dans les deux langues (catégories (1) et (2)) sont les plus rares.

Classe	(1)	(2)	(3-fr)	(3-en)	(4)
Pourcentage	5.4	5.5	16.1	15.2	57.8
Exemple	renouvelable	soviétique	francs	patrie	savon

TABLE 4 – Proportion des mots dans les différentes classes de comportement de dérive inter-langues, avec un exemple pour chaque classe.

Considérons à titre d'exemple le mot *Barbie*, qui appartient à la 3ème catégorie. L'espace aligné des plongements de mots est réduit à deux dimensions au moyen de la méthode t-SNE (van der Maaten & Hinton, 2008) pour représenter l'évolution de ce mot dans les deux langues (figure 3). Les mots les plus similaires au mot *Barbie* dans chaque langue, et à chaque strate temporelle, sont indiqués sur le graphique. En français (en rouge), le mot cible ne subit pas d'évolution notable. Il est majoritairement associé au criminel nazi *Klaus Barbie* et à son procès. Ces évènements ont eu une couverture médiatique moins importante et plus ponctuelle dans les journaux américains ; l'équivalent anglais du mot cible évolue rapidement en direction du champs lexical de la mode, en association avec la célèbre marque de poupée homonyme. Son plongement sémantique se stabilise dans ce voisinage à partir des années 2000.

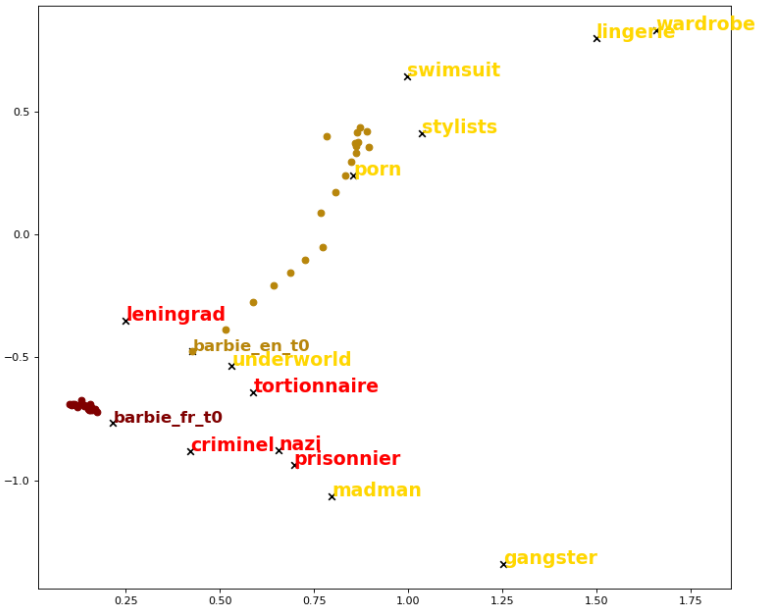

FIGURE 3 – T-SNE des espaces de plongements sémantiques alignés, représentant l'évolution temporelle du mot *Barbie* en français (rouge) et en anglais (jaune) ainsi que leurs plus proches voisins au cours du temps sur des strates annuelles.

5 Discussion

Dans cet article, nous étudions en détail le comportement d'un modèle d'apprentissage de plongements lexicaux dynamiques. Le point de départ de notre étude est le modèle *Dynamic Bernoulli Embeddings*, dont nous définissons plusieurs variantes. Nous répliquons ainsi le comportement d'autres modèles d'apprentissage de plongements de mots diachroniques de la littérature. Deux propriétés nous paraissent importantes à distinguer pour bien caractériser ces modèles : la capacité à mettre en évidence des évolutions dirigées des plongements de mots, et la capacité à garder une partie du vocabulaire stable au cours du temps. Nous montrons ensuite qu'il est possible d'analyser l'évolution d'un mot dans deux langues de façon conjointe. Un processus d'analyse préliminaire est mis en place en initialisant les modèles dynamiques à partir de plongements de mots alignés, et en analysant la dérive de la similarité inter-langues.

Le domaine en plein essor qu'est l'apprentissage de plongements de mots dynamiques manque encore de la cohésion que possèdent les tâches plus anciennes du traitement automatique des langues. Les publications sur ce sujet portent sur des corpus très diversifiés et les évaluations se font le plus souvent de façon qualitative, en l'absence de base d'évaluation robuste. Notons qu'un cadre d'évaluation est difficile à définir dans le cas qui nous intéresse, tant les attentes applicatives vis-à-vis d'un modèle diachronique peuvent varier. De plus, un cadre mathématique commun et rigoureux n'a pas encore été défini (Kutuzov *et al.*, 2018) et devrait s'appuyer sur les modèles d'apprentissage conjoint sur toutes les strates temporelles tel que celui décrit ici.

En effet, l'apprentissage conjoint à travers toutes les strates permet de s'affranchir dans une certaine mesure de la nécessité d'avoir un grand volume de données dans chacune d'elle. Néanmoins, le caractère discontinu des strates temporelles induit le modèle à détecter seulement les dérives d'une strate à l'autre ; plus les strates sont larges, plus les variations internes sont cachées, ou du moins moyennées. Ainsi, la question de la juste granularité temporelle se pose et dépend de l'application visée. Il est cependant important que les modèles étudiés puissent travailler à différents niveaux de finesse temporelle. Par exemple, lors de la recherche de dérives sémantiques brusques et de court terme, un type de modèle en temps continu (Rosenfeld & Erk, 2018) pourrait être plus adéquat, mais nécessite des informations temporelles très précises qui en pratique se retrouvent presque exclusivement dans les corpus issus de médias sociaux.

Une alternative est d'explorer l'usage de processus temporels de diffusion plus complexes, travaux initiés par exemple par (Bamler & Mandt, 2017) avec le processus d'Ornstein-Uhlenbeck. Enfin, l'emploi de strates temporelles non fixes dont les ruptures seraient apprises en même temps que les plongements lexicaux, assorti d'une régularisation sur la fonction de coût similaire à celle du modèle DBE-SC, serait une alternative à explorer. Néanmoins le cadre théorique approprié à cette sorte de quantification temporelle apprise par le modèle reste à définir.

Références

AITCHISON J. (2001). Language change : Progress or decay ? In *Cambridge Approaches to Linguistics*. Cambridge University Press.

BAMLER R. & MANDT S. (2017). Dynamic word embeddings. In D. PRECUP & Y. W. TEH, Eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings*

of Machine Learning Research, p. 380–389, International Convention Centre, Sydney, Australia : PMLR.

BENGIO Y., DUCHARME R., VINCENT P. & JAUVIN C. (2003). A neural probabilistic language model. In *Journal of Machine Learning Research*, p. 1137–1155.

CONNEAU A., LAMPLE G., RANZATO M., DENOYER L. & JÉGOU H. (2018). Word translation without parallel data. *CoRR*, **abs/1710.04087**.

DUBOSSARSKY H., WEINSHALL D. & GROSSMAN E. (2017). Outta control : Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 1136–1145 : Association for Computational Linguistics.

EGER S. & MEHLER A. (2016). On the linearity of semantic change : Investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 52–58, Berlin, Germany : Association for Computational Linguistics.

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

HAMILTON W. L., LESKOVEC J. & JURAFSKY D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1489–1501 : Association for Computational Linguistics.

HAN R., GILL M., SPIRLING A. & CHO K. (2018). Conditional word embedding and hypothesis testing via bayes-by-backprop. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 4890–4895 : Association for Computational Linguistics.

KIM Y., CHIU Y.-I., HANAKI K., HEGDE D. & PETROV S. (2014). Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, p. 61–65 : Association for Computational Linguistics.

KULKARNI V., AL-RFOU R., PEROZZI B. & SKIENA S. (2015). Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, p. 625–635, Republic and Canton of Geneva, Switzerland : International World Wide Web Conferences Steering Committee.

KUTUZOV A., ØVRELID L., SZYMANSKI T. & VELLDAL E. (2018). Diachronic word embeddings and semantic shifts : a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 1384–1397 : Association for Computational Linguistics.

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

ROSENFELD A. & ERK K. (2018). Deep neural models of semantic shift. In *NAACL 2018*.

RUDOLPH M. & BLEI D. (2018). Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, p. 1003–1011 : International World Wide Web Conferences Steering Committee.

RUDOLPH M., RUIZ F., MANDT S. & BLEI D. (2016). Exponential family embeddings. In *Advances in Neural Information Processing Systems*, p. 478–486.

SANDHAUS E. (2008). The new york times annotated corpus. In *Philadelphia : Linguistic Data Consortium*. Vol. 6, No. 12.

SZYMANSKI T. (2017). Temporal word analogies : Identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 448–453 : Association for Computational Linguistics.

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

TANG X. (2018). A state-of-the-art of semantic change computation. *Natural Language Engineering*, **24**(5), 649–676.

VAN DER MAATEN L. & HINTON G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605.

YAO Z., SUN Y., DING W., RAO N. & XIONG H. (2018). Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, p. 673–681 : ACM.

Apprentissage de plongements lexicaux par une approche réseaux complexes

Victor Connes[1, 2] Nicolas Dugué[2]
(1) Le Mans Université, LIUM, EA 4023, Laboratoire d'Informatique de l'Université du Mans
(2) LS2N Université de Nantes – faculté des Sciences et Techniques (FST) Bâtiment 34 2 Chemin de la Houssinière BP 92208, 44322 Nantes Cedex 3
victor.connes@gmail.com, nicolas.dugue@univ-lemans.fr

RÉSUMÉ

La littérature des réseaux complexes a montré la pertinence de l'étude de la langue sous forme de réseau pour différentes applications : désambiguïsation, résumé automatique, classification des langues, *etc.* Cette même littérature a démontré que les réseaux de co-occurrences de mots possèdent une structure de communautés latente. Nous formulons l'hypothèse que cette structuration du réseau sous forme de communautés est utile pour travailler sur la sémantique d'une langue et introduisons donc dans cet article une méthode d'apprentissage de plongements originale basée sur cette hypothèse. Cette hypothèse est cohérente avec la proximité qui existe entre la détection de communautés sur un réseau de co-occurrences et la factorisation d'une matrice de co-occurrences, méthode couramment utilisée pour l'apprentissage de plongements lexicaux. Nous décrivons notre méthode structurée en trois étapes : construction et pré-traitement du réseau, détection de la structure de communautés, construction des plongements de mots à partir de cette structure. Après avoir décrit cette nouvelle méthodologie, nous montrons la pertinence de notre approche avec des premiers résultats d'évaluation sur les tâches de catégorisation et de similarité. Enfin, nous discutons des perspectives importantes d'un tel modèle issu des réseaux complexes : les dimensions du modèle (les communautés) semblent interprétables, l'apprentissage est rapide, la construction d'un nouveau plongement est presque instantanée, et il est envisageable d'en expérimenter une version incrémentale pour travailler sur des corpus textuels temporels.

ABSTRACT

Complex networks based word embeddings.

Most of the time, the first step to learn word embeddings is to build a word co-occurrence matrix. As such matrices are equivalent to graphs, complex networks theory can naturally be used to deal with such data. In this paper, we consider applying community detection, a main tool of this field, to the co-occurrence matrix corresponding to a huge corpus. Community structure is used as a way to reduce the dimensionality of the initial space. Using this community structure, we propose a method to extract word embeddings that are comparable to the state-of-the-art approaches.

MOTS-CLÉS : Plongements lexicaux, réseaux complexes, détection de communautés.

KEYWORDS: Word embeddings, complex networks, community detection.

1 Introduction

Dans l'état de l'art de l'apprentissage de plongements lexicaux, on recense de nombreuses approches basées sur une matrice de co-occurrences termes-termes construite en utilisant de grands corpus (Pennington *et al.*, 2014; Levy *et al.*, 2015). Les auteurs factorisent ensuite cette matrice creuse de façon à obtenir un nouvel espace dans lequel chaque terme est représenté par un vecteur dense.

Dans le domaine des réseaux complexes, ces matrices de co-occurrences sont appelées *graphes* ou *réseaux*. L'étude du langage naturel par le prisme des réseaux complexes n'est pas une science nouvelle. L'état de l'art du domaine utilise également de grands corpus pour construire des réseaux $G = (V, E)$ tels que chaque nœud $u \in V$ du réseau représente un terme du vocabulaire, et un lien $(u, v) \in E$ entre deux nœuds représente une co-occurrence dans le corpus entre deux termes. Ces réseaux peuvent être dirigés, ou valués, on se dote alors d'une fonction w qui associe un poids à chaque lien $w : E \to \mathbb{R}$.

Ces travaux ont notamment permis de révéler plusieurs propriétés de ces réseaux et ainsi de mieux comprendre la façon dont est construite la langue : ces réseaux sont petit-monde (i Cancho & Solé, 2001), sans-échelle avec une loi de puissance à deux vitesses (i Cancho & Solé, 2001) expliquée par le modèle de Dorogovtsev & Mendes 2001, et le poids des liens suit également une loi de puissance dans le cas des réseaux valués (Gao *et al.*, 2014; Masucci & Rodgers, 2006).

Parmi les propriétés observées, ce papier se concentre sur la présence d'une structure de communautés dans ces réseaux (Newman, 2004). La structure de communautés d'un réseau est une partition des nœuds du réseau telle que pour chaque partie, les nœuds sont plus connectés entre eux qu'avec le reste du réseau (Newman & Girvan, 2004). Nous faisons l'hypothèse que cette structure de communautés permet de construire des plongements lexicaux.

Cette hypothèse se base sur deux constats. Le premier vient des exemples de Palla *et al.* 2005 qui semblent indiquer que les communautés encapsulent une partie de l'information sémantique. D'ailleurs, la définition de la structure de communautés vient appuyer ce constat : pour chaque partie (communauté) de la partition (structure de communautés), les nœuds sont plus connectés entre eux qu'avec le reste du réseau. Au regard de l'hypothèse de Firth "*a word is characterized by the company it keeps*", on comprend que chaque communauté sera constituée de mots qui seront utiles pour se caractériser les uns les autres. Le second constat vient de certains travaux de la littérature qui mettent en évidence les liens entre décomposition en valeur singulière (SVD) et détection de communautés (Sarkar & Dong, 2011). Or, appliquer une SVD à une matrice de co-occurrences pondérée par la *positive pointwise mutual information* est une méthode efficace pour aboutir à des plongements lexicaux (Levy *et al.*, 2015).

Nous présentons donc Section 2 notre approche basée sur la détection de communautés pour extraire des plongements. Cette approche considère chaque communauté comme une dimension, et les liens d'un nœud vers ces communautés permettent de calculer pour chaque dimension la valeur de la composante. Nous montrons Section 3 que les résultats expérimentaux démontrent la pertinence de l'approche, d'un point de vue qualitatif, mais également quantitatif. Enfin, nous discuterons Section 4 des avantages d'une telle approche. Tout d'abord, celle-ci permet d'espérer des dimensions interprétables. Ensuite, le calcul d'un plongement pour un terme est très rapide. Enfin, ce type d'approche ouvre des perspectives pour créer des plongements lexicaux évoluant dans le temps via des algorithmes de détection de communautés incrémentaux.

2 Méthode

Données. Les données utilisées sont les GoogleBooksNgram [1] anglais, corpus BristishEnglish et EnglishFiction. Les GoogleBooksNgram sont des recueils de co-occurrences de termes observées sur une grande bibliothèque de textes allant des années 1800 à 2008. Les co-occurrences sont fournies avec une fenêtre de contexte allant de 2 à 5. Pour nos expériences, nous avons conservé seulement les co-occurrences observées depuis 1980 aboutissant à un vocabulaire avant pré-traitements d'environ 380000 termes.

Construction et pré-traitement du réseau. Une fois le réseau créé en exploitant les co-occurrences d'un corpus textuel avec une fenêtre de taille f, on obtient alors $G = (V, E, w)$ comme décrit. Pour rappel, l'ensemble des nœuds V est équivalent au *vocabulaire* considéré, l'ensemble des liens E représente les co-occurrences entre les termes du vocabulaire, et on définit la pondération des liens de E avec la fonction $w(u, v)$, qui vaut le nombre de co-occurrences observées entre les termes représentés par les nœuds u et v dans le corpus en considérant le paramètre f.

Dans le but de ne conserver que les co-occurrences ayant une valeur sémantique, nous supprimons les liens entre les nœuds qui ne révèlent pas une dépendance statistique significative en utilisant l'Éq. 1 :

$$ppmi(w, c) = max\left(0, log_2\left(\frac{p(u, v)}{p(v)p(u)}\right)\right) \quad (1)$$

Ce pré-traitement du réseau découle directement de ce qui est préconisé par l'état de l'art, notamment par Levy *et al.* 2015. Mais il semble également pertinent de l'appliquer pour simplifier le travail de l'algorithme de détection de communautés, dont les résultats s'améliorent avec des pré-traitements de type seuillage ou repondération (Yan *et al.*, 2018).

Dans le but d'alléger le réseau avec un filtre basse-fréquence, nous appliquons l'algorithme 1 qui permet d'obtenir le k-cœur du réseau (Matula & Beck, 1983) : il s'agit de supprimer tous les nœuds ayant moins de k voisins de manière récursive jusqu'à que tous les nœuds restant dans le réseau soient connectés à au moins k voisins.

Enfin, dans le but de supprimer les mots vides et de limiter l'influence des hautes fréquences (comme dans Glove (Pennington *et al.*, 2014) ou Word2vec (Mikolov *et al.*, 2013), nous choisissons de supprimer les $ntop$ nœuds de plus haut degré. Les meilleurs résultats sont empiriquement obtenus pour $ntop$ = 200 et k = 10. Après pré-traitement, nous aboutissons à un vocabulaire de 135.000 mots dans le cas de notre corpus.

Détection de communautés. Une fois le réseau généré et pré-traité, la seconde étape consiste à détecter les communautés qui serviront par la suite de dimensions aux vecteurs de plongement lexicaux. On dit que C est une partition de V telle que $C = \{C_0, C_1, ..., C_n\}$ avec $\cup_i C_i \in C = V$. S'il n'existe pas de définition unique du concept de communauté, une structure de communautés est souvent définie comme une partition du réseau telle que les nœuds de chaque partie sont plus connectés entre eux qu'avec le reste du réseau.

1. http ://storage.googleapis.com/books/ngrams/books/datasetsv2.html

Algorithm 1 Extraction du K-cœur

Require: $G = (V, E)$ graphe, k entier
  $convergence \leftarrow False$
  while $convergence$ **do**
    $convergence \leftarrow True$; $V' \leftarrow \{\}$
    for $\forall n \in V$ **do**
      if $degres(n) < k$ **then**
        $V' \leftarrow V' \cup \{n\}$; $convergence \leftarrow False$
      end if
    end for
    $V \leftarrow V \setminus V'$
  end while
  return G

De nombreuses méthodes existent pour réaliser l'extraction de ces communautés. Nous avons choisi l'algorithme 2 de propagation de labels introduit par Raghavan *et al.* 2007, dont la complexité est quasi-linéaire en $O(|V|)$, et qui génère *théoriquement* des communautés dont les tailles suivent une distribution permettant d'éviter d'avoir en grand nombre des communautés trop grandes (fourre-tout) ou trop petites (trop spécifiques) (Dao *et al.*, 2018). En pratique, on constate Figure 1 un très grand nombre de petites communautés. Il s'agirait de considérer des adaptations de l'algorithme de propagation de labels pour éviter cet écueil.

Algorithm 2 Propagation de labels

Require: $G = (V, E, w)$ graphe
  $\forall n \in V, c(n) \leftarrow n$
  On parle de convergence lorsque la communauté de chaque nœud est la communauté majoritaire de ses voisins.
  while $check\_convergence(G, C, w)$ **do**
    for $\forall n \in V in\ random\ order$ **do**
      La communauté du nœud devient la communauté majoritaire des voisins.
      $C(n) \leftarrow countmax(\{c(v),\ \forall v \in voisins(n)\}, w)$
    end for
  end while
  return C

Extraction des plongements lexicaux. Une fois les communautés extraites, il reste à construire les plongements pour notre vocabulaire. Pour ce faire, nous considérons la distribution des liens de chaque nœud à travers les communautés. Néanmoins, il s'agit de prendre en compte l'influence du degré du nœud et de celui de ses voisins. Prenons un exemple pour clarifier : celui des mots *escroc* et *aigrefin*. Ces deux mots sont proches d'un point de vue sémantique. Par contre, *escroc* est plus fréquent qu'*aigrefin*. Il sera donc mécaniquement d'un degré pondéré plus élevé. Une fois cette remarque faite, on se rend compte que si on considère seulement la distribution d'*escroc* dans les communautés pour créer son plongement, la norme de son vecteur sera plus grande que celle d'*aigrefin*.

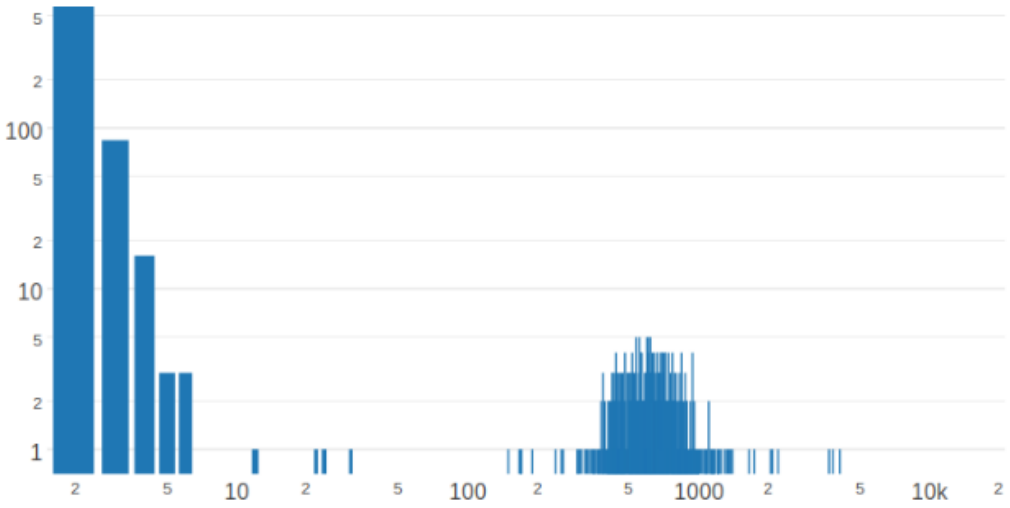

FIGURE 1 – Distribution de la taille des communautés (en log-log)

Par ailleurs, la taille des communautés a une influence similaire. L'algorithme de détection de communautés aboutit (sauf exception) à une partition dont les tailles des communautés sont hétérogènes. Si l'on ne tient pas compte de cet état de fait, les communautés les plus petites auront mécaniquement une composante plus faible que les grosses communautés dans les vecteurs.

Ainsi, si on note e_n le plongement du nœud représentant le mot n, $e_n \in \mathbb{R}^{|C|}$ et e_n^c la valeur de la composante correspondante à la communauté c de e_n, cette valeur se calculera ainsi :

$$\hat{e}_n^c = \frac{1}{|N^c(e_n)|} \sum_{v \in N^c(e_n)} sppmi(n, v) \quad (2) \qquad e_n^c = \frac{\hat{e}_n^c - \mu(\hat{e}_*^c)}{\sigma(\hat{e}_*^c)} \quad (3)$$

Avec $N^c(e_n) = voisins(n) \cap C_c$, *i.e.* le nombre de voisins du noeud représentant le mot n appartenant à la communauté c, $\mu(\hat{e}_*^c)$ et $\sigma(\hat{e}_*^c)$ respectivement la moyenne et l'écart-type des valeurs de $\hat{e}_n^c$, $\forall n \in V$ et $sppmi$ une version normalisée de la $ppmi$ (Éq. 1) à valeur dans $[0, 1]$.

L'utilisation de la $sppmi$ nous permet de contrebalancer l'influence du degré du nœud et de celui de ses voisins, celle du *z-score* (Éq. 3) l'influence de la taille des communautés. L'exemple de la Figure 2 illustre le résultat une fois toutes les étapes réalisées, en proposant une visualisation des vecteurs de *bush*, *putin* et *chirac* via les 30 dimensions les plus utiles pour la caractériser (10 par vecteur).

3 Résultats

Nous débuterons cette section avec quelques évaluations empiriques purement qualitatives concernant la pertinence des dimensions exploitées (les communautés) et l'espace appris (les voisinages). Nous donnerons enfin des résultats quantitatifs qui démontrent l'intérêt de l'approche. Sur notre corpus, après pré-traitement, nous aboutissons à une taille de vocabulaire qui est d'un peu plus de 135.000 mots, ce qui correspond au nombre de nœuds du graphe. Après détection de communautés, nous obtenons environ 30.000 communautés (voir la distribution de leurs tailles Figure 1), soit des vecteurs de taille 30.000. En revanche, on constate qu'en moyenne, seulement 300 (environ) composantes du vecteur sont non-nulles, les vecteurs sont donc extrêmement creux.

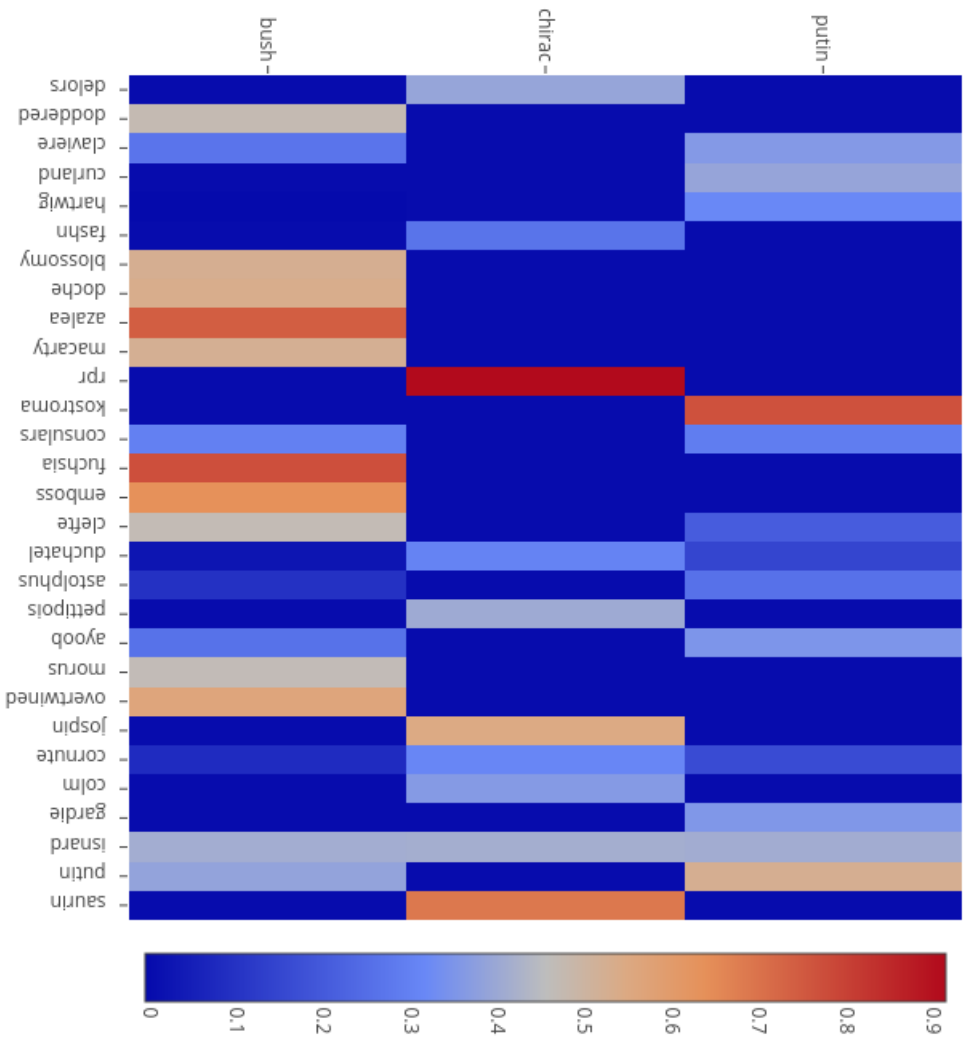

FIGURE 2 – En abscisse, les étiquettes des 30 dimensions les plus caractéristiques des vecteurs de putin, chirac et bush (10 par vecteur). La couleur représente la valeur de la composante pour chacun des vecteurs.

Communautés et interprétabilité. En utilisant des méthodes d'étiquetage des communautés, il est possible d'évaluer empiriquement la pertinence de l'approche. Les communautés extraites constituent les dimensions des vecteurs qui semblent ainsi interprétables et cohérentes. Nous donnons ici en exemple trois communautés et leurs étiquettes caractéristiques :

— ('officiel', 'republique', 'parisienne', 'couture', 'senat')
— ('copper', 'iron', 'stand', 'metal', 'upon')
— ('volleyball', 'handball', 'softball', 'badminton', 'basketball')

La première communauté regroupe les mots français du corpus. La seconde concentre du vocabulaire lié aux métaux même si l'on peut constater qu'on y trouve des intrus (*stand* et *upon*). Enfin, la dernière communauté regroupe du vocabulaire lié au sport. Ce sont les mêmes méthodes d'étiquetage qui sont utilisées dans la Figure 2 pour étiqueter chaque dimension des vecteurs visualisés. On reconnaît un bon nombre de ces étiquettes comme par exemple *Delors* (Jacques), *rpr* (Rassemblement pour la république), *Jospin* (Lionel) pour caractériser le vecteur de Jacques Chirac. Il est particulièrement intéressant de s'intéresser au vecteur *bush*, où l'on trouve *Mcarthy* (Joseph), ou encore *Putin* (Vladimir) mais également des noms d'arbustes puisque c'est l'un des sens de *bush* (*azalea*, *fuschia*). Grâce à l'interprétabilité du modèle, nous pouvons ainsi observer la façon dont celui-ci intègre la polysémie/l'homonymie.

Base canonique et interprétabilité. Avec une approche telle que celle de *Word2Vec*, il est difficile d'interpréter les dimensions. Tout d'abord, supposer qu'il est possible d'interpréter les dimensions revient à faire l'hypothèse que chaque dimension peut être considérée indépendamment des autres, et que chaque dimension a un sens cohérent, i.e. qu'explorer les vecteurs colinéaires aux vecteurs de la base canonique de l'espace appris permettrait d'extraire ces *sens*. Empiriquement, il est pourtant difficile d'affirmer cela. Considérons un modèle Word2vec à 300 dimensions appris sur le corpus Google News (Mikolov *et al.*, 2013), et prenons des contre-exemples simples. Soit C la base canonique de l'espace de dimension 300 de notre expérimentation telle que $C = \{e_1 = (1, 0, 0, \cdots, 0), e_2 = (0, 1, 0, \cdots, 0), \cdots, e_{300} = (0, 0, 0, \cdots, 1)\}$.

— Considérons ainsi les 10 termes les plus proches de e_4 dans l'espace : Ginsburgs, Dinty Moore, jelly sandwiches, cheartier appetites, they'd, Fabens fliers, banana republics, isn, Chipotle burritos, payroll deduction.
— Pour e_9, nous obtenons les résultats suivants : costliest natural disasters, counterparty defaults, mute button, closely scrutinized, Bernankes, damage Minsch, student Tyler Clementi, degraded Kenneth Merten, historian Bob Kreipke, Nishu Sood.

Il semble très difficile de tirer une quelconque cohérence dans les termes qui sont retournés, contrairement aux communautés précédemment citées. Les communautés sont en effet des objets concrets, des ensembles de mots du corpus qui sont particulièrement connectés ensemble, et elles peuvent de plus être considérées indépendamment les unes des autres.

Considérons maintenant notre modèle et les vecteurs canoniques de l'espace constitué via l'extraction des communautés. Dans notre modèle comme dans les autres, il est possible d'extraire les plus proches voisins d'un mot ou d'un vecteur en utilisant la similarité *cosine* pour évaluer la distance entre deux vecteurs. Considérons dans les cas des deux exemples suivants les 10 vecteurs les plus proches du vecteur canonique dont la composante non-nulle correspond à la communauté qui contient le mot **alcohol**, puis **petal** :

— mannite, dinitro, polyhydric, benzole, benzol, lactose, fermenter, disaccharide, bisulphide, reconverted.
— sepal, papilionaceous, floret, stamen, blotch, dewdrops, petals, bracts, corolla.

Dans le premier exemple, de manière générale, on obtient des termes liés à l'alcool directement : *mannite* pour le manniotal qui est un alcool, *polyhydric* parce que les alcools de sucre sont dits polyhydriques ; des termes liés au sucre qui est l'un des éléments de base pour la création d'alcool (*lactose*, *disaccharise*), au processus de création d'alcool (*fermenter*), ou à la chimie (*dinitro*, *bisulphide*). Dans le second exemple, tout ou presque est lié à la fleur : les sépales (*sepal*), les étamines (*stamen*), *papilionaceous* qui est une fleur, *drewdrop* qui signifie goutte de rosée, *bract* qui est une petite feuille, *corolla* qui est un synonyme de pétale. Dans ces deux cas, il existe un fort recouvrement entre les 10 plus proches voisins mentionnés ci-dessus et les représentants les plus caractéristiques de la communauté (étiquettes).

Un autre exemple parlant est celui du vecteur e_{746} de notre modèle dont les 10 plus proches voisins sont : sifteen, fiftyfour, fortyseven, fiftyseven, sixtyfive, fortysix, fiftyfive, sixtyseven, twentyeight, twentyseven.

La distance *cosine* peut comme dans les autres modèles être exploitée pour étudier la similarité entre les termes du vocabulaire, pas seulement avec les vecteurs canoniques. Ainsi, la liste de voisins suivante fournit quelques exemples de résultats illustrant le bon fonctionnement de la méthode :

— metal : (metals, metallic, iron, copper, steel, alloy, aluminium, oxides, chromium)
— picture : (pictures, portrait, image, painting, view, images, depiction, portrayal, painted)
— salad : (mayonnaise, ketchup, lettuce, tomato, vegetables, sauce, celery, mashed, cheese)
— mars : (altimeter, orbiter, venus, saturn, jupiter, orbit, pioneer, planets, planet)
— news : (television, cnn, bbc, pathe, nbc, tidings, newspapers, cbs, gaumont)

Comparaison à l'état de l'art. Pour obtenir des résultats quantitatifs, nous comparons notre approche à celles de l'état de l'art en considérant deux tâches d'évaluation (Schnabel *et al.*, 2015) :

Similarité La tâche de similarité se présente comme une base de données de paires de mots, avec pour chaque paire un score associé. Le score de similarité entre deux mots est issu d'une évaluation humaine. La qualité du modèle peut donc être évaluée en calculant la corrélation entre le vecteur de score humain et le vecteur de distances entre les vecteurs appris. Une corrélation linéaire (coefficient de *Spearman* proche de 1) correspond à un modèle complètement en accord avec l'évaluation humaine.

Catégorisation La tâche de catégorisation se présente comme une base de données de paires (mot, catégorie). Le but est de réussir à regrouper des mots en différentes catégories en utilisant les vecteurs appris. Pour faire cela, on opère une analyse de regroupement sur les vecteurs appris. On évalue ensuite le modèle en calculant la pureté entre les regroupements et la catégorisation humaine.

Nous utilisons la librairie *word-embeddings-benchmarks* [2] pour réaliser nos évaluations (Jastrzebski *et al.*, 2017). Nous comparons nos résultats à ceux obtenus avec des plongements pré-entraînés accessibles en ligne en utilisant cette librairie. Les plongements utilisés sont ceux obtenus via les méthodes Glove (Pennington *et al.*, 2014), NMT (Hill *et al.*, 2014), HDC et PDC (Sun *et al.*, 2015), Skip-gram (Mikolov *et al.*, 2013) et Lexvec (Salle *et al.*, 2016).

Les résultats Table 1 sont encourageants, ils montrent que notre approche est pertinente. Pour chaque tableau, nous comparons les résultats de notre méthode aux meilleurs résultats des méthodes de l'état de l'art citées, pour 50 et 300 dimensions. Sur deux corpus (en gras dans la Table), l'un exploité

2. https ://github.com/kudkudak/word-embeddings-benchmarks

Benchmark	Notre modèle	État de l'art dim=300	État de l'art dim=50
		Similarité	
MEN	0.650	0.809	0.720
SimLex	0.364	0.427	0.309
RG65	**0.803**	0.790	0.763
		Catégorisation	
ESSLI1a	0.75	0.818	0.773
ESSLI2b	**0.775**	0.750	0.775
ESSLI2c	0.6	0.667	0.556

TABLE 1 – Résultats sur les tâches de catégorisation et de similarité en comparaison de l'état de l'art.

pour la tâche de similarité, l'autre pour la tâche de catégorisation, notre méthode obtient des résultats comparables à celles de l'état de l'art auxquelles nous nous comparons. Dans le reste des cas, notre méthode obtient des résultats supérieurs aux performances des approches de l'état de l'art paramétrés pour retourner des vecteurs en dimension 50, mais inférieurs lorsque ces vecteurs sont en dimension 300. En accord avec l'état de l'art nos meilleurs résultats sont obtenus pour les plus grandes tailles de fenêtre ($f = 5$ dans notre cas).

4 Discussion et perspectives

Nous avons décrit une méthode originale d'apprentissage de plongements lexicaux basée sur une approche réseaux complexes. Nous proposons d'utiliser les communautés détectées sur le réseau de co-occurrences représentant le corpus comme dimensions de nos plongements. Les vecteurs sont ensuite directement extraits de la distribution des liens de chaque nœud à travers la structure communautaire. Les résultats qualitatifs et quantitatifs montrent la pertinence de l'approche qui obtient des scores comparables à l'état de l'art. Néanmoins, une étude avec les mêmes méta-paramètres (corpus, taille de fenêtre) semblent nécessaire pour se situer exactement par rapport à l'état de l'art.

Cette approche a pour avantage de fournir des dimensions qui sont des objets concrets, physiquement existants : les communautés. Ces dimensions semblent donc interprétables : il est possible de consulter le contenu de ces communautés, de les étiqueter avec des éléments caractéristiques. Néanmoins, cela ne garantit pas l'interprétabilité des vecteurs appris. Pour que ces vecteurs soient interprétables, il s'agit à notre sens de réunir deux conditions. La première, est de disposer d'un étiquetage suffisamment précis pour qu'il soit tout à fait compréhensible. La seconde nécessite d'avoir un vecteur de taille raisonnable, ou du moins un vecteur creux afin de ne pas avoir trop de communautés à inspecter. Ces questions sont en lien direct avec le paramétrage des algorithmes de détection de communautés et constituent des perspectives directes de notre travail. Nous souhaitons en effet travailler à évaluer l'interprétabilité des vecteurs extraits par notre méthode par des humains.

De plus, notre méthode permet l'extraction rapide du plongement d'un mot ou d'une expression. Le calcul de ce vecteur découle en effet directement de la connectivité du nœud qui représente le mot, de la façon dont ses liens se dispersent au sein de la structure communautaire. Ainsi, le calcul du vecteur d'un nouveau mot ou d'une expression composée ne nécessite pas de réapprendre un modèle, mais

simplement d'ajouter le terme au réseau pour extraire le vecteur.

Enfin, les langues évoluent avec le temps : le sens des mots change ou de nouveaux sens apparaissent. Ces évolutions de la langue ont été décrites, notamment par Bloomfield 1983; Mitra *et al.* 2015. Des travaux considèrent des méthodes automatiques basées sur les plongements lexicaux et de grands corpus temporels pour la détection de ces néologismes sémantiques (Tang, 2018). Ces méthodes peuvent être séparées en deux classes. La première classe est celle des méthodes *diachroniques* : elles discrétisent le temps et séparent ainsi le corpus en plusieurs sous-corpus. Sur chacun de ces sous-corpus, les auteurs proposent d'apprendre des plongements lexicaux puis d'aligner les espaces appris entre les sous-corpus deux à deux (Hamilton *et al.*, 2016). Ces approches sont basées sur l'hypothèse très forte qu'il est possible d'aligner des espaces différents issus d'algorithmes non déterministes aboutissant à des résultats sous-optimaux. La seconde classe, celle des méthodes *dynamiques*, propose une optimisation globale de tous les plongements du vocabulaire à travers le temps, aboutissant à un problème gourmand en calcul et très difficile (Bamler & Mandt, 2017). Notre approche peut permettre d'ouvrir le champ à de nouveaux travaux basés sur les algorithmes de détection de communautés incrémentaux (Xie *et al.*, 2013). Cela permettrait ainsi de s'abstraire d'une optimisation globale coûteuse, et de contourner l'hypothèse d'alignement diachronique des espaces.

Références

BAMLER R. & MANDT S. (2017). Dynamic word embeddings. In *International Conference on Machine Learning*, p. 380–389.

BLOOMFIELD L. (1983). *An introduction to the study of language*, volume 3. John Benjamins Publishing.

DAO V.-L., BOTHOREL C. & LENCA P. (2018). Estimating the similarity of community detection methods based on cluster size distribution. In *International Workshop on Complex Networks and their Applications*, p. 183–194 : Springer.

DOROGOVTSEV S. N. & MENDES J. F. F. (2001). Language as an evolving word web. *Proceedings of the Royal Society B : Biological Sciences*, **268**(1485), 2603–2606.

GAO Y., LIANG W., SHI Y. & HUANG Q. (2014). Comparison of directed and weighted co-occurrence networks of six languages. *Physica A : Statistical Mechanics and its Applications*, **393**, 579–589.

HAMILTON W. L., LESKOVEC J. & JURAFSKY D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv :1605.09096*.

HILL F., CHO K., JEAN S., DEVIN C. & BENGIO Y. (2014). Embedding word similarity with neural machine translation. *arXiv preprint arXiv :1412.6448*.

I CANCHO R. F. & SOLÉ R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London B*, **268**(1482), 2261–2265.

JASTRZEBSKI S., LEŚNIAK D. & CZARNECKI W. M. (2017). How to evaluate word embeddings ? on importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv :1702.02170*.

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

MASUCCI A. P. & RODGERS G. J. (2006). Network properties of written human language. *Physical Review E*, **74**(2), 026102.

MATULA D. W. & BECK L. L. (1983). Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, **30**(3), 417–427.

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.

MITRA S., MITRA R., MAITY S. K., RIEDL M., BIEMANN C., GOYAL P. & MUKHERJEE A. (2015). An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering*, **21**(5), 773–798.

NEWMAN M. E. (2004). Analysis of weighted networks. *Physical review E*, **70**(5), 056131.

NEWMAN M. E. & GIRVAN M. (2004). Finding and evaluating community structure in networks. *Physical review E*, **69**(2), 026113.

PALLA G., DERÉNYI I., FARKAS I. & VICSEK T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, **435**(7043), 814–818.

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.

RAGHAVAN U. N., ALBERT R. & KUMARA S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, **76**(3).

SALLE A., IDIART M. & VILLAVICENCIO A. (2016). Enhancing the lexvec distributed word representation model using positional contexts and external memory. *arXiv preprint arXiv :1606.01283*.

SARKAR S. & DONG A. (2011). Community detection in graphs using singular value decomposition. *Physical Review E*, **83**(4).

SCHNABEL T., LABUTOV I., MIMNO D. & JOACHIMS T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 298–307.

SUN F., GUO J., LAN Y., XU J. & CHENG X. (2015). Learning word representations by jointly modeling syntagmatic and paradigmatic relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, volume 1, p. 136–145.

TANG X. (2018). A State-of-the-Art of Semantic Change Computation. *arXiv preprint arXiv :1801.09872*.

XIE J., CHEN M. & SZYMANSKI B. K. (2013). Labelrankt : Incremental community detection in dynamic networks via label propagation. In *Proceedings of the Workshop on Dynamic Networks Management and Mining*, p. 25–32 : ACM.

YAN X., JEUB L. G., FLAMMINI A., RADICCHI F. & FORTUNATO S. (2018). Weight thresholding on complex networks. *arXiv preprint arXiv :1806.07479*.

Comparaison qualitative et extrinsèque d'analyseurs syntaxiques du français : confrontation de modèles distributionnels sur un corpus spécialisé

Ludovic Tanguy[1] Pauline Brunet[2] Olivier Ferret[2]
(1) CLLE-ERSS : CNRS & Université de Toulouse, France
(2) CEA LIST, Laboratoire Analyse Sémantique Texte et Image, Gif-sur-Yvette, F-91191 France.
ludovic.tanguy@univ-tlse2.fr, {pauline.brunet,olivier.ferret}@cea.fr

RÉSUMÉ

Nous présentons une étude visant à comparer 11 différents analyseurs en dépendances du français sur un corpus spécialisé (constitué des archives des articles de la conférence TALN). En l'absence de gold standard, nous utilisons chacune des sorties de ces analyseurs pour construire des thésaurus distributionnels en utilisant une méthode à base de fréquence. Nous comparons ces 11 thésaurus afin de proposer un premier aperçu de l'impact du choix d'un analyseur par rapport à un autre.

ABSTRACT

Extrinsic evaluation of French dependency parsers on a specialised corpus : comparison of distributional thesauri

We present a study with the goal of comparing 11 different french dependency parsers applied on a specialised corpus (consisting of articles from the archives of the TALN conference). Due to the lack of a gold standard, we use each of the parsers' output to generate distributional thesauri using a frequency-based method. We compare these 11 thesauri in order to propose a first look at the impact of choosing to use a parser over another.

MOTS-CLÉS : analyse syntaxique, analyse distributionnelle, domaine de spécialité, évaluation.

KEYWORDS: French dependency parsing, distributional semantics, specialised corpus.

1 Introduction

Cet article se place dans le cadre d'une étude des méthodes de sémantique distributionnelle en domaine spécialisé et en français. Notre objectif à moyen terme est la sélection de la méthode la plus efficace pour identifier des similarités sémantiques distributionnelles entre les unités lexicales ou terminologiques d'un petit corpus de langue spécialisée (quelques millions de mots au plus). On sait que de très nombreux paramètres interviennent et doivent faire l'objet de choix éclairés, sans que l'on puisse malheureusement se fonder sur des conclusions obtenues sur de grands corpus génériques. Nous faisons tout d'abord l'hypothèse qu'une méthode à base de contextes syntaxiques permet de compenser la faible quantité de données, ce qui a en partie été montré par (Tanguy *et al.*, 2015). La question que nous étudions plus précisément ici est l'impact du choix d'un analyseur syntaxique en amont d'une méthode fréquentielle de construction d'un modèle distributionnel. Ce faisant, nous rejoignons les questions de la comparaison de l'efficacité des différents outils et modèles disponibles

pour l'analyse syntaxique en dépendances.

Il y a eu de nombreux efforts dans la communauté du Traitement Automatique des Langues (TAL) du français pour comparer les différents analyseurs syntaxiques : campagnes Easy (Paroubek *et al.*, 2008), Passage (De La Clergerie *et al.*, 2008), SPMRL (Seddah *et al.*, 2013), CoNLL (Zeman *et al.*, 2018) ou des comparaisons plus ponctuelles comme celles de (Candito *et al.*, 2010) ou (De La Clergerie, 2014). Mais les bancs de test utilisés ne sont pas forcément pertinents pour l'analyse de corpus spécialisés, préférant les corpus génériques et variés sur lesquels les outils sont entraînés. De plus, malgré les campagnes récentes, les principaux outils disponibles ne sont pas nécessairement tous comparés sur les mêmes jeux d'évaluation.

En l'absence d'étalon adapté à nos besoins, nous avons donc procédé à une comparaison qualitative en confrontant les principaux analyseurs actuels du français sur une tâche externe, à la manière des récentes campagnes EPE pour l'anglais (Fares *et al.*, 2018). Ne disposant pour cette tâche que d'un jeu de test de couverture et de validité limitées, l'essentiel de notre évaluation reste purement comparative. Nous examinerons donc avant tout l'ampleur et la portée des modifications apportées par le changement d'analyseur syntaxique sur les thésaurus distributionnels obtenus en fin de chaîne.

Notre dispositif est donc le suivant : sur un même corpus spécialisé en français, nous avons appliqué 11 analyseurs syntaxiques en dépendances (ou versions) et extrait pour chacun d'eux les principaux contextes syntaxiques que nous avons utilisés pour construire autant de modèles distributionnels en utilisant une méthode classique à base de fréquence. Nous comparons au final les thésaurus distributionnels obtenus afin d'identifier à la fois l'impact réel de l'analyseur sur la chaîne mais nous proposons aussi un début de cartographie des analyseurs en fonction de la similarité des modèles qu'ils ont permis de générer.

Dans la section 2, nous présentons le corpus choisi et les différents analyseurs (ou configurations d'analyseurs) qui y ont été appliqués. Puis nous présentons en section 3 les différentes étapes de normalisation et de sélection des sorties de ces analyseurs que nous avons dû appliquer pour obtenir un socle commun de comparaison ainsi qu'une première comparaison fondée sur ce socle. La section 4 présente le processus de construction des modèles distributionnels et leur comparaison suivant différentes approches.

2 Matériau et outils testés

2.1 Corpus TALN

Nous avons utilisé pour cette expérimentation un corpus spécialisé de petite taille : le corpus TALN [1], constitué des archives des actes des conférences TALN et RECITAL des années 1999 à 2014 incluses. Ce corpus d'environ 4,5 millions de mots possède plusieurs avantages pour l'étude des modèles d'analyse distributionnelle : il s'agit de textes écrits de bonne qualité, homogènes en termes de genre et de thématique et relevant d'un thème pour lequel nous avons un niveau d'expertise suffisant pour pouvoir interpréter facilement les résultats d'une analyse distributionnelle.

La conversion en texte brut depuis les fichiers PDF initiaux a été réalisée avec le seul objectif de disposer de texte compatible avec un analyseur syntaxique, au détriment d'un ensemble d'éléments comme les titres de section, notes de bas de page, tableaux, formules mathématiques, références

1. Disponible sur `http://redac.univ-tlse2.fr/corpus/taln.html`

bibliographiques et autres légendes. Les césures ont été éliminées et l'ensemble du texte ainsi filtré de chaque article est présenté sur une même ligne. La robustesse des analyseurs face à des données trop bruitées n'a donc pas été un paramètre de cette étude.

2.2 Analyseurs testés

Nous avons sélectionné 7 outils proposant une analyse syntaxique du français en dépendances, en nous concentrant sur ceux disponibles facilement et prêts à l'emploi (i.e. prenant en charge l'ensemble de la chaîne de traitement du texte brut jusqu'aux dépendances syntaxiques). Ces outils ont été appliqués avec toutes les options par défaut.

Tous ces outils se fondent sur des modèles obtenus par apprentissage sur des corpus annotés. La multiplicité des analyseurs est en fait essentiellement due à des choix d'implémentation concernant les techniques d'analyse en dépendances (par graphe ou par transitions par exemple), les modèles d'apprentissage (modèles classiques type SVM ou entropie maximale ou plus récemment réseaux de neurones récurrents) et les traitements en amont ou périphériques (segmentation, lemmatisation). Les corpus d'entraînement sont, eux, bien plus réduits en nombre, ce qui s'explique bien entendu par le coût élevé du processus d'annotation et de validation. Avant de présenter les outils que nous avons étudiés, il est donc primordial de rappeler les corpus disponibles, puisque ceux-ci ont un impact décisif sur la nature et le format des sorties.

FTB Le premier corpus annoté syntaxiquement pour le français est le *Corpus arboré du français*, plus connu sous son nom anglais de *French Treebank* ou *FTB* (Abeillé *et al.*, 2003). Constitué d'environ 600 000 mots issus du journal Le Monde, il a tout d'abord été annoté en suivant une analyse en constituants, puis converti automatiquement en dépendances par (Candito *et al.*, 2010). Une autre version a également été produite pour la campagne d'évaluation SPMRL, notamment pour le repérage des unités polylexicales (Seddah *et al.*, 2013).

UD French Afin de faciliter le développement et la comparaison des différents analyseurs ainsi que des études typologiques crosslingues à grande échelle, un schéma universel de dépendances a été proposé, fondé sur le modèle des *Stanford Dependencies*, désormais baptisé *Universal Dependencies* (ou UD)[2] (Nivre *et al.*, 2016). Le projet UD propose des lignes génériques ainsi que des jeux d'étiquettes universels (pour les catégories grammaticales et les relations syntaxiques) et a permis de regrouper sous un même format et de diffuser différents corpus annotés du français, notamment :

- **UD French FTB** est la conversion en UD du French Treebank original et contient environ 550 000 mots.
- **UD French ParTUT** est la partie française du corpus multilingue Parallel-TUT (Bosco *et al.*, 2012), composé d'échantillons de textes variés (textes de lois, entrées de Wikipédia, pages Facebook, etc.) pour un total d'environ 30 000 mots.
- **UD French GSD** est le corpus initial du projet UD qui prend ses sources dans les dépendances de Stanford (McDonald *et al.*, 2013). Il contient 400 000 mots issus d'articles de journaux, de pages Wikipédia, de blogs ou de critiques de divers produits.
- **UD French Sequoia** est un corpus développé en complément du FTB dans le but (entre autres) d'en améliorer la couverture en genre et en domaines (Candito & Seddah, 2012).

2. Voir `http://universaldependencies.org/` pour un historique détaillé et les différentes versions de son développement.

> Les 70 000 mots de ce corpus sont issus de débats parlementaires, de presse régionale, de Wikipédia et de textes médicaux. Initialement annoté suivant le schéma du FTB, il a été converti au format UD.

Il est à noter que malgré la volonté de normalisation et d'universalisation du projet UD, les différents corpus cités précédemment n'utilisent pas exactement les mêmes conventions ni les mêmes jeux d'étiquettes pour les relations syntaxiques. Ces différences s'expliquent par des positions théoriques ou techniques face à certains phénomènes syntaxiques, mais aussi par les différentes étapes de conversion qu'ont connues certains corpus.

En ce qui concerne les analyseurs, nous avons sélectionnés 7 outils différents, dont certains proposent des variantes en termes de modèles pré-entraînés, autrement dit de corpus.

CoreNLP (Manning *et al.*, 2014), l'analyseur principal de l'équipe de Stanford, implémente un étiqueteur à entropie maximale et un analyseur syntaxique par transitions. Il a été entraîné sur le corpus UD GSD.

StanfordNLP (Qi *et al.*, 2018) est un outil qui, en plus de permettre d'accéder aux fonctionnalités de CoreNLP en Python, implémente une chaîne de traitement neuronale entièrement différente. Son analyseur syntaxique par graphes repose sur un réseau neuronal LSTM. StanfordNLP propose plusieurs modèles pour le français. Nous en avons utilisé deux, entraînés respectivement sur les corpus UD **GSD** et **Sequoia**.

NLPCube (Boroş *et al.*, 2018) est, comme StanfordNLP, fondé sur des réseaux de neurones récurrents LSTM. Sa particularité principale est une analyse syntaxique indépendante de l'étiquetage morphosyntaxique, l'une comme l'autre utilisant uniquement des attributs lexicalisés, sans information morphologique. Il est significativement plus lent que tous les autres outils utilisés. Nous n'avons pas trouvé d'indication précise concernant les corpus sur lesquels le modèle fourni a été entraîné et nous avons présumé que la somme des corpus UD disponibles pour le français a été utilisée.

Spacy est un outil à visée industrielle qui met en avant sa rapidité par rapport aux autres outils disponibles. L'étiqueteur est un perceptron moyenné avec des attributs liés aux clusters de Brown suivant Koo *et al.* (2008). Il implémente un analyseur syntaxique par transitions non-monotone qui peut revenir sur des décisions antérieures (Honnibal & Johnson, 2015). Le modèle fourni a été entraîné sur le corpus WikiNER (Nothman *et al.*, 2012) pour la reconnaissance d'entités nommées et sur UD Sequoia pour l'étiquetage et l'analyse syntaxique.

UDPipe (Straka & Straková, 2017) accomplit la tokenisation et la segmentation dans un même temps avec un réseau de neurones récurrent à portes (GRU). Pour l'étiquetage, il génère des étiquettes possibles à partir du suffixe du mot puis désambiguïse à l'aide d'un perceptron moyenné. L'analyse syntaxique par transitions est fondée sur un réseau neuronal simple à une couche. UDPipe propose plusieurs modèles pour le français. Nous en avons utilisé trois, entraînés respectivement sur les corpus UD **GSD**, **Sequoia** et **ParTUT**.

Talismane (Urieli & Tanguy, 2013) utilise une combinaison de modèles statistiques aux traits spécifiques à chaque langue et de règles incorporant de la connaissance linguistique. En plus de la version distribuée entraînée sur le French TreeBank converti en dépendances (**FTB**), nous avons également utilisé une version expérimentale utilisant un modèle Universal Dependencies (**UD**) entraîné sur la concaténation de tous les corpus UD décrits précédemment.

MSTParser (McDonald *et al.*, 2006) est un analyseur en dépendances par graphes. Nous l'avons

utilisé à l'aide du paquet BONSAI [3], qui le couple à l'étiqueteur MElt (Denis & Sagot, 2009) et met en œuvre le meilleur modèle MST selon le benchmark de (Candito *et al.*, 2010). Il s'appuie sur la version non-UD du FTB.

Nous sommes bien conscients que les analyseurs décrits ci-dessus ne sont comparables que sur un plan purement pratique puisqu'ils relèvent de technologies, de degrés de finalisation voire d'époques très différents et qu'ils sont fondés sur des données d'entraînement elles aussi non comparables. Néanmoins, ils forment une bonne partie du paysage actuel des solutions disponibles en termes d'analyse syntaxique robuste du français et sont à ce titre tous susceptibles d'être considérés.

3 Exploitation des sorties

La comparaison des 11 outils ou versions sélectionnés nécessite d'identifier ou de construire un terrain commun entre les différentes analyses produites. Plusieurs problèmes se posent en termes d'hétérogénéité des sorties : l'identification des unités, l'alignement des jeux d'étiquettes morphosyntaxiques, la lemmatisation et bien entendu les relations de dépendance syntaxique. Nous avons décidé de nous limiter aux mots simples relevant des classes ouvertes (noms, verbes, adjectifs et adverbes), sous leur forme lemmatisée et en leur associant leur catégorie. Pour les relations de dépendance, nous avons sélectionné les principales à la fois représentées dans tous les formalismes et jugées les plus utiles pour l'analyse distributionnelle.

3.1 Identification des mots

Notre étude, comme c'est le cas pour l'essentiel des travaux actuels en sémantique distributionnelle, porte sur des mots simples et suppose donc que la segmentation se fait de façon homogène par les différents analyseurs.

La question de la lemmatisation est cruciale pour le français et les petits corpus puisqu'elle permet de limiter la dispersion des unités et donc de lutter contre le manque de données. Elle permet également un lien (en amont ou en aval de l'analyse distributionnelle) avec des ressources lexicales ou terminologiques qui représentent généralement leurs entrées sous leur forme canonique. Mais la lemmatisation reste une opération délicate, que les analyseurs traitent de façon inégale. Dans notre liste de systèmes, notons deux cas particuliers : CoreNLP, qui ne propose pas de lemmatisation pour le français, et Spacy, qui semble ne pas prendre en compte la catégorie morphosyntaxique des mots avant de calculer leur lemme et propose donc des sorties majoritairement erronées (nous avons remarqué que toute forme fléchie pouvant correspondre à un verbe sera toujours lemmatisée en utilisant l'infinitif de ce verbe, même si cette forme a été correctement identifiée comme un nom ou un adjectif). Les autres analyseurs qui effectuent une lemmatisation standard peuvent prendre ponctuellement des décisions différentes pour certaines situations (lemmes inconnus absents, certains noms féminins lemmatisés au masculin, lemmes ambigus marqués comme tels, etc.).

Nous avons donc décidé de lemmatiser toutes les sorties avec le même outil en utilisant un lexique flexionnel de référence et en nous fondant sur la catégorie morphosyntaxique attribuée par l'analyseur et ce, pour chaque mot des classes ouvertes. Le lexique est constitué de la fusion de Morphalou (Romary *et al.*, 2004) et de Lefff (Sagot, 2010). En cas d'absence de la forme de surface du mot dans

3. `http://alpage.inria.fr/statgram/frdep/fr_stat_dep_mst.html`

le lexique, nous avons appliqué une stratégie de lemmatisation par substitution de la partie droite de la forme fondée sur le plus long suffixe trouvé dans le lexique en appliquant la méthode décrite dans (Tanguy & Hathout, 2007, p. 302). Ainsi, un mot inconnu du lexique comme *relemmatisons* catégorisé comme verbe sera lemmatisé en *relemmatiser* par analogie avec des couples (forme, lemme) dont la finale est commune comme *(schématisons, schématiser)*. Cette méthode robuste permet de traiter l'ensemble des cas de façon homogène et déterministe.

Dans toute la suite, les mots sont représentés par leur lemme ainsi calculé et leur catégorie grammaticale. Nous avons identifié 5 580 mots de classe ouverte ayant une fréquence minimale de 5 dans chacune des 11 sorties. Notre comparaison s'appuiera sur cet ensemble.

3.2 Extraction des triplets syntaxiques

L'étape suivante consiste à extraire les relations de dépendance entre deux mots qui serviront de représentation des contextes des mots pour l'analyse distributionnelle, suivant une longue tradition (Lin, 1998; Bourigault, 2002; Padó & Lapata, 2007; Baroni & Lenci, 2010; Lapesa & Evert, 2017). Dans tous ces travaux, le mode le plus classique de représentation des contextes d'apparition des mots est l'utilisation d'un triplet syntaxique du type (mot_dépendant, relation, mot_gouverneur). Par exemple de la phrase *"Nous avons utilisé un analyseur syntaxique"* (correctement analysée), on peut extraire les triplets *(analyseur, obj, utiliser)* et *(syntaxique, mod, analyseur)*. Ces triplets permettent en fait de produire chacun deux représentations contextuelles : une pour le dépendant, l'autre pour le gouverneur (avec une relation inversée), que l'on représente sous la forme de couples *(analyseur, utiliser_obj)*, *(utiliser, analyseur_obj-1)*, *(syntaxique, analyseur_mod)* et *(analyseur, syntaxique_mod-1)*. C'est sur la base de ces couples que le rapprochement des mots s'effectuera en appliquant le principe de l'analyse distributionnelle.

Il existe un grand nombre de possibilités pour générer ces triplets (et les couples correspondants) à partir des sorties d'un analyseur en dépendances. (Baroni & Lenci, 2010) ou (Lapesa & Evert, 2017) en ont proposé de nombreuses variantes, qui dépendent par exemple des relations syntaxiques considérées, du nombre de liens de dépendance suivis et de l'inclusion de certains mots (e.g. les prépositions) dans les relations syntaxiques. (Tanguy *et al.*, 2015) ont, sur le même corpus TALN, et en utilisant un jeu d'évaluation réduit, confirmé les conclusions de (Padó & Lapata, 2007) sur l'intérêt de se limiter à un jeu réduit de relations de dépendance, ce que nous avons fait ici.

Comme indiqué en section 2, les analyseurs testés sont entraînés sur des corpus différents et leurs sorties héritent donc des choix différents faits lors des campagnes d'annotation manuelle (ou d'une conversion automatique le cas échéant). Les différences principales dans notre cas sont celles existant entre les modèles issus du FTB et ceux des corpus de la famille UD, comme le montre la figure 1 pour une même phrase de notre corpus, correctement analysée par deux outils différents.

Cet exemple illustre que la normalisation des triplets nécessite de prendre en compte à la fois les différences de segmentation pour le traitement des articles contractés (*du/de le*), d'étiquettes des catégories morphosyntaxiques (N/NOUN, V/VERB), de liens de dépendance (advmod/mod) mais aussi dans la façon dont une même relation est traduite par plusieurs dépendances, comme c'est le cas pour le rattachement de *corpus* à *totalité* via la préposition *de*.

Nous avons au final choisi d'extraire les triplets correspondant aux relations suivantes :

`N suj V` : sujet nominal d'un verbe ;

`N obj V` : objet direct nominal d'un verbe ;

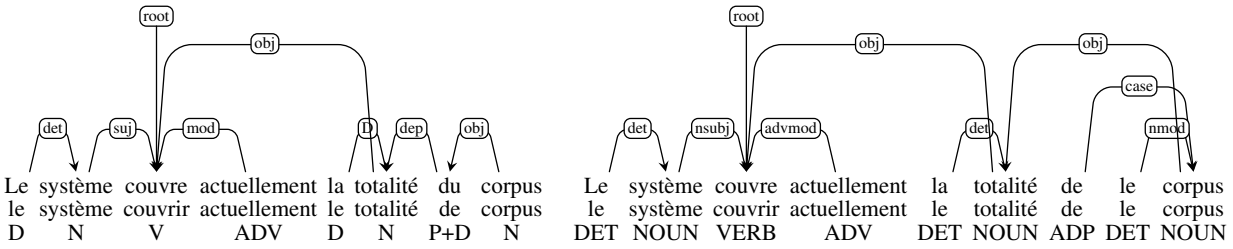

FIGURE 1 – Dépendances identifiées pour la phrase "Le système couvre actuellement la totalité du corpus" par MSTParser (à gauche) et UDPipe-Sequoia (à droite)

`ADJ mod N` : adjectif modifieur d'un nom ou attribut du sujet nominal ;

`ADV mod ADJ/V` : adverbe modifieur d'un adjectif ou d'un verbe ;

`X coord X` : coordination entre deux noms, verbes, adverbes ou adjectifs ;

`X prep_P X` : rattachement prépositionnel entre nom, verbe ou adjectif.

Dans ce dernier cas, nous avons intégré la préposition à la relation, si bien que le cas ci-dessus de "totalité du corpus" donne les triplets (*totalité, prep_de, corpus*) et (*corpus, prep_de-1, totalité*).

Nous avons également normalisé les locutions prépositionnelles et adverbiales dont certaines sont identifiées par certains des analyseurs. Talismane-FTB, par exemple, identifiera "à partir de" comme une préposition complexe directement au niveau de la segmentation (*à_partir_de*) alors que les analyseurs UD comme NLPCube utiliserons une relation de dépendance spécifique (fixed) qui rattache *de* à *partir* et *partir* à *à*. Dans les deux cas, nous avons reconstruit la préposition complexe et ses relations de dépendance externes en utilisant la première notation. Nous avons de plus exclus explicitement les cas où le verbe est un modal et ceux où l'adverbe est une négation.

Ces extractions ont nécessité le développement de règles spécifiques pour s'adapter à chaque famille de format de sortie. Ceci a pu conduire à regrouper certaines relations et assimiler des distinctions que certains formats d'annotation feraient et pas d'autres.

3.3 Comparaison des triplets syntaxiques

Le nombre de triplets (occurrences) extraits est assez stable et va de 2,13 millions pour Spacy à 2,67 millions pour Talismane-UD. Les triplets uniques (types) vont de 1,04 million pour Spacy à 1,32 million pour UDPIPE-Partut. Les triplets impliquant un mot du vocabulaire commun (cf. ci-dessus) rassemblent un total de 2,8 millions de triplets différents (dont seuls 10%, soit 261 965, ont été repérés par tous les analyseurs). La comparaison des accords entre les analyseurs sur les fréquences des triplets donne une corrélation (Spearman) moyenne de 0,49. On peut observer plus précisément certaines tendances dans les rapprochements opérés sur cette base au niveau de la figure 2.

On remarquera que ce sont les jeux d'étiquettes (UD *vs* FTB) qui semblent avoir l'impact le plus important, comme attendu, avec un isolement de MSTParser et de Talismane-FTB. Il y a en revanche de très importantes variations au sein des analyseurs entraînés sur les corpus de la famille UD, sans que l'on puisse à ce stade identifier un rôle prédominant de l'architecture ou du corpus d'entraînement.

Nous avons examiné manuellement les différences en parcourant la liste des triplets ayant une fréquence importante dans l'une des sorties et absents d'une ou plusieurs autres. Les principaux phénomènes que nous avons pu identifier à la source de ces désaccords sont les suivants :

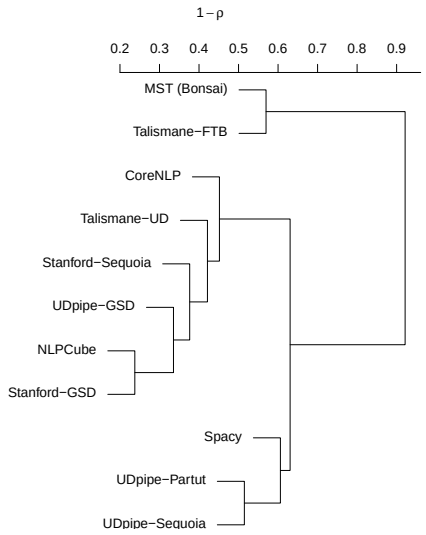

FIGURE 2 – Classification hiérarchique ascendante des analyseurs en fonction de leur corrélation sur les fréquences des triplets repérés par au moins deux analyseurs différents

— segmentation (ou non) des mots composés (*mot-cible, hors-contexte*, etc.) ;
— étiquetage de certains mots : *même* (ADJ, ADV, PRON), *tout* (ADJ, ADV, DET, N, PRON), *certain* (ADJ, DET, PRON), numéraux (ADJ, NUM, N) ;
— prise en compte des majuscules lors de l'étiquetage ; le cas de désaccord le plus courant est *TA* (pour traduction automatique) : N ou DET ;
— repérage de locutions (*d'abord*, *à partir de*, *par exemple* etc.), même en prenant en compte les différentes stratégies comme indiqué plus haut ;
— catégorisation (et donc lemmatisation) des participes (présents et passés : ADJ ou V) ;
— catégorisation des composés N-N (*candidat terme*, *langue cible*, *vecteur contexte* : étiquetés comme N-N, ADJ-N, N-ADJ ou autre).

Sans chercher à harmoniser ces différents points, nous avons utilisé ces jeux de triplets pour construire des modèles distributionnels.

4 Comparaison des modèles distributionnels

4.1 Construction des modèles

Pour reprendre la distinction faite dans (Baroni *et al.*, 2014), la construction des contextes distributionnels dans cette étude s'inscrit dans une tradition d'approches à base de comptes (Lin, 1998) par opposition aux approches prédictives (Mikolov *et al.*, 2013). Ce choix est doublement motivé. Tout d'abord, l'utilisation de relations de dépendance dans les approches prédictives reste rare, à l'exception de (Levy & Goldberg, 2014). Plus fondamentalement, un certain nombre de travaux récents (Pierrejean & Tanguy, 2018) ont montré que les approches prédictives se caractérisent par une certaine instabilité du point de la recherche des plus proches voisins. Afin de nous concentrer sur les différences résultant de la seule utilisation de différents analyseurs syntaxiques, nous avons donc opté pour une approche à base de comptes.

Sa mise en œuvre est très classique et reprend les acquis d'un certain nombre d'études récentes (Kiela

& Clark, 2014; Baroni *et al.*, 2014; Levy *et al.*, 2015), et plus particulièrement de (Ferret, 2010) au travers de deux principaux éléments : l'adoption de l'information mutuelle ponctuelle positive pour pondérer les couples (cooccurrent, relation) au sein des contextes distributionnels et l'application d'un filtrage très limité se contentant de supprimer les couples n'ayant qu'une seule occurrence dans les contextes. Ce dernier choix est motivé à la fois par la taille restreinte du corpus d'étude et les expérimentations réalisées dans (Ferret, 2010) dans le cas des cooccurrents linéaires.

Nous avons calculé la similarité entre deux mots d'un modèle en utilisant la mesure classique du cosinus sur chaque paire de mots pour laquelle cela était possible (i.e. chaque paire de mots ayant au moins un contexte commun).

4.2 Comparaison globale des modèles

Nous avons tout d'abord comparé chaque modèle aux autres en calculant le coefficient de corrélation de Spearman sur les scores de similarité (cosinus) sur l'ensemble des paires de mots du vocabulaire commun. Comme précédemment, nous avons synthétisé cette comparaison en faisant une analyse hiérarchique ascendante à la figure 3. On y retrouve, comme précédemment, que trois modèles sont identifiés comme plus atypiques (MST, Talismane-FTB et Spacy), les autres étant très regroupés.

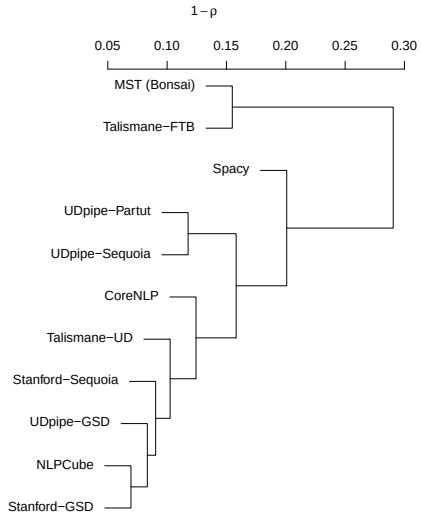

FIGURE 3 – Classification hiérarchique ascendante des modèles en fonction de leur accord sur les cosinus des paires de mots communs

Nous avons ensuite calculé l'accord entre les modèles concernant les premiers voisins proposés pour chaque mot par les différents modèles. Parmi les mots du vocabulaire commun décrit précédemment, seuls 4 469 avaient au moins un voisin distributionnel dans ce même ensemble. Nous avons calculé pour chaque paire de modèles le taux d'accord relatif sur le premier voisin et utilisé cette similarité pour faire une analyse hiérarchique ascendante présentée à la figure 4. On peut y voir une organisation différente de celle de la figure 2, bien que Spacy, MST et Talismane-FTB y restent les plus excentriques.

En étendant la comparaison des plus proches voisins aux 25 premiers voisins, nous reprenons la méthode utilisée par (Pierrejean & Tanguy, 2018) pour mesurer la stabilité des méthodes neuronales d'analyse distributionnelle. Le taux moyen est de 0,58, ce qui signifie que seuls 42% des 25 premiers voisins se retrouvent (en moyenne) d'un modèle à un autre. Le score de variation moyen peut

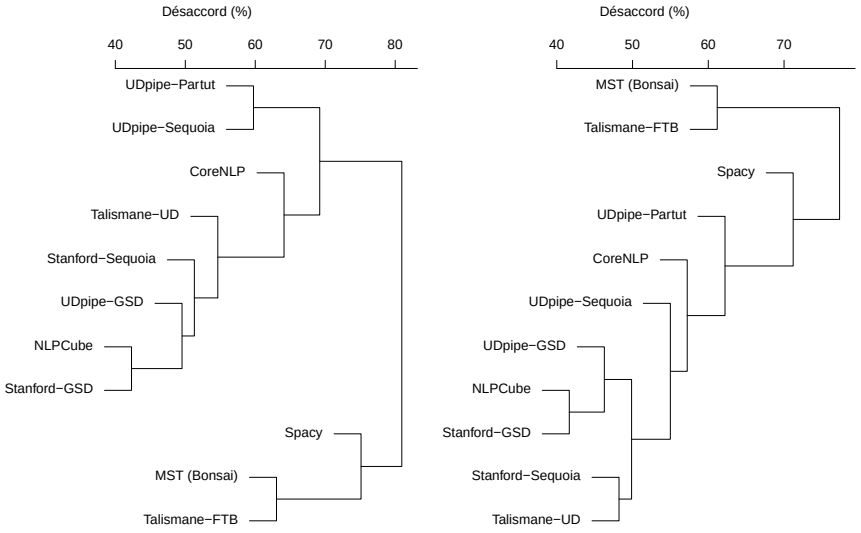

FIGURE 4 – Classification hiérarchique ascendante des modèles en fonction de leur accord sur le plus proche voisin (à gauche) et sur les 25 plus proches voisins (à droite)

être utilisé pour comparer deux à deux chacun des modèles, ce que nous avons synthétisé par une classification hiérarchique ascendante à la figure 4 (à droite). On y voit une modification par rapport à la prise en compte du seul premier voisin et au final, un rapprochement des résultats obtenus avec la corrélation sur les cosinus en figure 3. Là encore, les trois analyseurs les plus atypiques sont Spacy, MST et Talismane-FTB, tandis qu'un noyau dur des analyseurs fondés sur UD est formé par Stanford, NLPCube et UDPipe-GSD.

Ces grandes tendances se retrouvent au niveau de la comparaison des voisins des différents thésaurus considérés réalisée grâce à la mesure *Rank-Biased Overlap* (Webber *et al.*, 2010) et illustrée par la figure 5. Cette mesure est appliquée à tous les voisins (100) des entrées extraits pour chaque thésaurus et étend la notion de recouvrement moyen – la moyenne du recouvrement entre deux listes pour les différents rangs de ces listes – afin de donner une importance décroissante aux recouvrements à mesure de l'augmentation des rangs, donnant ainsi un poids plus important aux premiers voisins. Cette importance est fixée au travers du paramètre p, vu comme la probabilité de poursuivre le parcours des listes comparées après chaque item depuis leur début. Ainsi, la valeur $p = 0,98$ utilisée ici revient à dire que les 50 premiers voisins concentrent de l'ordre de 85% du poids de la mesure. La figure 5 est obtenue grâce à la distance 1 - RBO, qui a les propriétés d'une métrique.

D'un point de vue qualitatif, nous avons observé 322 mots pour lesquels les 11 modèles sont unanimes concernant leur plus proche voisin. Nous avons pu identifier différents cas de figure parmi ceux-ci :

— des antonymes ou synonymes (dans le domaine TAL) de haute fréquence : *sortie/entrée*, *qualité/performance*, *ordonner/trier*, *valeur/score*, *texte/document* etc.
— des antonymes/synonymes de basse fréquence : *empiler/dépiler*, *intimement/étroitement*, *expérimentalement/empiriquement*, *mélodique/intonatif*, *itérer/réitérer*
— des paires accidentelles que l'on peut expliquer par des contextes exclusifs et systématiques : *parti/leçon* (*tirer_obj*) *routier/hydraulique* (*barrage_mod-1*), *adjacence/covariance* (*matrice_prep_de*), *metteur/mettre* (*scène_prep_en-1*)

À l'inverse, on trouve 10 cas de désaccord total (un plus proche voisin différent pour chaque modèle),

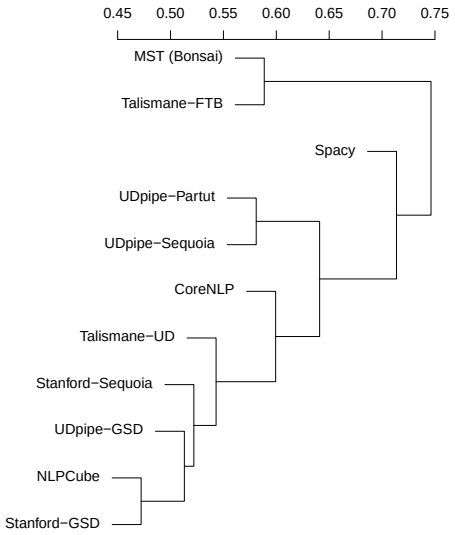

FIGURE 5 – Classification hiérarchique ascendante des modèles en fonction de la mesure RBO

tous de basse fréquence. Dans certains de ces cas, plusieurs des plus proches voisins peuvent être considérés comme pertinents, comme pour *auxiliaire*, où l'on retrouve majoritairement différentes notions grammaticales : *adverbe, déterminant, copule, croisé, gérondif, numéral, transitif, fraction, économiser, laps, subordination*. Dans d'autres cas, le bruit domine nettement, comme pour *post* : *subtilité, multi, bilan, jeudi, billet, chargement, délocaliser, syllabation, SRI, mercredi, pool*, où seul *billet* peut être considéré comme pertinent.

4.3 Évaluation sur un jeu de test *ad hoc*

Nous avons enfin voulu avoir un aperçu, même limité et partiel, des performances relatives de ces différents analyseurs, en évaluant leur capacité à identifier des similarités pertinentes pour le domaine du TAL à partir du corpus. Dans une étude visant à comparer l'impact des différents paramètres gouvernant la construction d'un modèle distributionnel, Tanguy *et al.* (2015) avaient développé un petit jeu d'évaluation sur le corpus TALN en faisant évaluer par quatre juges experts du domaine la pertinence des voisins proposés par un ensemble de systèmes sur 15 mots pivots sélectionnés.

Le jeu de données [4] contient ainsi, pour 5 verbes (*annoter, calculer, décrire, extraire, évaluer*), 5 noms (*fréquence, graphe, méthode, sémantique, trait*) et 5 adjectifs (*complexe, correct, important, précis, spécialisé*) une série de mots déclarés similaires par les juges, avec comme score synthétique pour chaque voisin d'un mot-cible le nombre de juges l'ayant retenu. Par exemple, pour le nom *trait*, on y trouve les mots suivants avec leur score : *attribut* (4), *caractéristique* (4), *propriété* (4), *étiquette* (4), *catégorie* (3), *descripteur* (2), *feature* (2), *indice* (2), *information* (2) ... *marque* (1), *représentation* (1), *structure* (1).

Le jeu est partiel puisque, pour chacun des pivots, seuls les mots ayant été ramenés comme un des trois plus proches voisins par un des systèmes initialement considérés ont été évalués (720 configurations différentes au total). Il est donc possible que des voisins pertinents proposés par un de nos 11 systèmes n'aient pas été considérés ; mais nous supposerons ici que le cas est marginal.

Sur la base de ce jeu de test, nous avons donc calculé pour chaque pivot et pour chaque modèle la

4. Disponible ici : `http://redac.univ-tlse2.fr/datasets/semdis-gold/TAL56-2/`

Modèle	Rang moyen
Talismane-UD	2,38
MST (Bonsai)	2,88
Talismane-FTB	3,00
Stanford-Seq	4,25
UDPipe-Seq	5,75
CubeNLP	6,13
UDpipe-Partut	6,25
UDPipe-GSD	6,38
CoreNLP	7,50
Stanford-GSD	9,38
Spacy	11,00

TABLE 1 – Rang moyen des analyseurs sur le jeu de test de (Tanguy *et al.*, 2015) fondé sur leur score cumulé aux rangs (1, 5, 10, 15, 20, 25, 50, 100)

somme des scores de pertinence des plus proches voisins à différents rangs (1, 5, 10, 15, 20, 25, 50 et 100). L'ordre entre les modèles variant peu au final, nous reportons en table 1 leur rang moyen sur ces différentes positions.

L'ordonnancement des outils laisse voir des extrémités très marquées, dans lesquelles on retrouve en fait les analyseurs les plus excentriques des comparaisons précédentes. Spacy semble celui qui produit les résultats les moins en accord avec l'annotation initiale alors que le trio constitué de MST/Bonsai et des deux versions de Talismane semble se détacher. On remarquera également que si pour Stanford le choix du corpus d'entraînement est critique, ce n'est pas le cas pour UDPipe.

5 Conclusion et perspectives

Les variations que nous avons mesurées entre des modèles sémantiques ne différant que par les analyseurs syntaxiques utilisés sont très importantes : le choix d'un analyseur syntaxique n'est certainement pas anodin pour l'analyse distributionnelle de petits corpus spécialisés. Les différences observées en fin de chaîne, lorsque l'on compare les voisins distributionnels, ne sont pas nécessairement corrélées à celles que l'on trouve en comparant directement les sorties. On a toutefois pu observer que tous les mots, contextes et paires de mots similaires ne sont pas tous affectés de la même façon par le changement d'analyseur. Cette première étape aura néanmoins permis d'identifier, parmi les différents outils testés, ceux dont les comportements sont les plus différents. Des sondages plus pointus et des examens manuels des cas de désaccord devraient nous permettre d'identifier les contextes syntaxiques les plus critiques, et par là même nous aider à choisir l'analyseur le plus adapté dans ce dispositif.

Remerciements

Le travail présenté dans cet article a été réalisé dans le cadre du projet ADDICTE [5] (Analyse distributionnelle en domaine spécialisé), financé par l'Agence Nationale de la Recherche (ANR-17-CE23-0001). Les auteurs tiennent en outre à remercier tous les membres partenaires du projet ADDICTE, et plus particulièrement Nabil Hathout pour son aide pour la relemmatisation des sorties.

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

Références

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). Building a treebank for French. In *Treebanks*, p. 165–187. Springer.

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

BARONI M. & LENCI A. (2010). Distributional memory : A general framework for corpus-based semantics. *Computational Linguistics*, **36**(4), 673–721.

BOROŞ T., DUMITRESCU S. D. & BURTICA R. (2018). NLP-cube : End-to-end raw text processing with neural networks. In *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 171–179, Brussels, Belgium : Association for Computational Linguistics.

BOSCO C., SANGUINETTI M. & LESMO L. (2012). The parallel-TUT : a multilingual and multiformat treebank. In *Proceedings of LREC*, p. 1932–1938 : European Language Resources Association (ELRA).

BOURIGAULT D. (2002). UPERY : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. In J.-M. PIERREL, Ed., *Actes de TALN 2002 (Traitement automatique des langues naturelles)*, p. 75–84, Nancy : ATALA ATILF.

CANDITO M., NIVRE J., DENIS P. & ANGUIANO E. H. (2010). Benchmarking of statistical dependency parsers for French. In *Proceedings of the 23rd International Conference on Computational Linguistics : Posters*, p. 108–116 : Association for Computational Linguistics.

CANDITO M. & SEDDAH D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Actes de TALN*, p. 321–334.

DE LA CLERGERIE É. V. (2014). Jouer avec des analyseurs syntaxiques. In *Actes de TALN*.

DE LA CLERGERIE E. V., HAMON O., MOSTEFA D., AYACHE C., PAROUBEK P. & VILNAT A. (2008). Passage : from French parser evaluation to large sized treebank. In *Proceedings of LREC*.

DENIS P. & SAGOT B. (2009). Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*.

FARES M., OEPEN S., ØVRELID L., BJ J., JOHANSSON R. *et al.* (2018). The 2018 shared task on extrinsic parser evaluation : On the downstream utility of english universal dependency parsers. *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 22–33.

FERRET O. (2010). Similarité sémantique et extraction de synonymes à partir de corpus. In *17ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2010)*, Montréal, Canada.

HONNIBAL M. & JOHNSON M. (2015). An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1373–1378, Lisbon, Portugal : Association for Computational Linguistics.

KIELA D. & CLARK S. (2014). A Systematic Study of Semantic Vector Space Model Parameters. In *2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, p. 21–30, Gothenburg, Sweden.

KOO T., CARRERAS X. & COLLINS M. (2008). Simple semi-supervised dependency parsing. *Proceedings of ACL-08 : HLT*, p. 595–603.

LAPESA G. & EVERT S. (2017). Large-scale evaluation of dependency-based DSMs : Are they worth the effort ? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, volume 2, p. 394–400.

LEVY O. & GOLDBERG Y. (2014). Dependency-Based Word Embeddings. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, p. 302–308, Baltimore, Maryland.

LEVY O., GOLDBERG Y. & DAGAN I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, **3**, 211–225.

LIN D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, p. 768–774 : Association for Computational Linguistics.

MANNING C. D., SURDEANU M., BAUER J., FINKEL J., BETHARD S. J. & MCCLOSKY D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, p. 55–60.

MCDONALD R., LERMAN K. & PEREIRA F. (2006). Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, p. 216–220 : Association for Computational Linguistics.

MCDONALD R., NIVRE J., QUIRMBACH-BRUNDAGE Y., GOLDBERG Y., DAS D., GANCHEV K., HALL K., PETROV S., ZHANG H., TÄCKSTRÖM O. *et al.* (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 92–97.

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

NIVRE J., DE MARNEFFE M.-C., GINTER F., GOLDBERG Y., HAJIC J., MANNING C. D., MCDONALD R. T., PETROV S., PYYSALO S., SILVEIRA N. *et al.* (2016). Universal dependencies v1 : A multilingual treebank collection. In *Proceedings of LREC*.

NOTHMAN J., RINGLAND N., RADFORD W., MURPHY T. & CURRAN J. R. (2012). Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, **194**, 151–175.

PADÓ S. & LAPATA M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, **33**(2), 161–199.

PAROUBEK P., ROBBA I., VILNAT A. & AYACHE C. (2008). EASY, Evaluation of Parsers of French : what are the results ? In *Proceedings of LREC*.

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.

QI P., DOZAT T., ZHANG Y. & MANNING C. D. (2018). Universal Dependency Parsing from Scratch. In *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 160–170, Brussels, Belgium : Association for Computational Linguistics.

ROMARY L., SALMON-ALT S. & FRANCOPOULO G. (2004). Standards going concrete : from LMF to Morphalou. In *COLING 2004 Enhancing and using electronic dictionaries*, p. 22–28, Geneva, Switzerland : COLING.

SAGOT B. (2010). The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In *Proceedings of LREC*, Valletta, Malta : European Languages Resources Association (ELRA).

SEDDAH D., TSARFATY R., KÜBLER S., CANDITO M., CHOI J. D., FARKAS R., FOSTER J., GOENAGA I., GOJENOLA GALLETEBEITIA K., GOLDBERG Y., GREEN S., HABASH N., KUHLMANN M., MAIER W., NIVRE J., PRZEPIÓRKOWSKI A., ROTH R., SEEKER W., VERSLEY Y., VINCZE V., WOLIŃSKI M., WRÓBLEWSKA A. & VILLEMONTE DE LA CLERGERIE E. (2013). Overview of the SPMRL 2013 shared task : A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, p. 146–182, Seattle, Washington, USA : Association for Computational Linguistics.

STRAKA M. & STRAKOVÁ J. (2017). Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 88–99, Vancouver, Canada : Association for Computational Linguistics.

TANGUY L. & HATHOUT N. (2007). *Perl pour les linguistes*. Hermès.

TANGUY L., SAJOUS F. & HATHOUT N. (2015). Évaluation sur mesure de modèles distributionnels sur un corpus spécialisé : comparaison des approches par contextes syntaxiques et par fenêtres graphiques. *Traitement automatique des langues*, **56**(2).

URIELI A. & TANGUY L. (2013). L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions : études de cas avec l'analyseur Talismane. In *Actes de TALN*, p. 188–201, Les Sables d'Olonne, France.

WEBBER W., MOFFAT A. & ZOBEL J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, **28**(4), 1–38.

ZEMAN D., HAJI J., POPEL M., POTTHAST M., STRAKA M., GINTER F., NIVRE J. & PETROV S. (2018). CoNLL 2018 shared task : Multilingual parsing from raw text to universal dependencies. *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 1–21.

Compression de vocabulaire de sens grâce aux relations sémantiques pour la désambiguïsation lexicale

Loïc Vial Benjamin Lecouteux Didier Schwab
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
{loic.vial, benjamin.lecouteux, didier.schwab}@univ-grenoble-alpes.fr

RÉSUMÉ

En Désambiguïsation Lexicale (DL), les systèmes supervisés dominent largement les campagnes d'évaluation. La performance et la couverture de ces systèmes sont cependant rapidement limités par la faible quantité de corpus annotés en sens disponibles. Dans cet article, nous présentons deux nouvelles méthodes qui visent à résoudre ce problème en exploitant les relations sémantiques entre les sens tels que la synonymie, l'hyperonymie et l'hyponymie, afin de compresser le vocabulaire de sens de WordNet, et ainsi réduire le nombre d'étiquettes différentes nécessaires pour pouvoir désambiguïser tous les mots de la base lexicale. Nos méthodes permettent de réduire considérablement la taille des modèles de DL neuronaux, avec l'avantage d'améliorer leur couverture sans données supplémentaires, et sans impacter leur précision. En plus de nos méthodes, nous présentons un système de DL qui tire parti des récents travaux sur les représentations vectorielles de mots contextualisées, afin d'obtenir des résultats qui surpassent largement l'état de l'art sur toutes les tâches d'évaluation de la DL.

ABSTRACT

Sense Vocabulary Compression through Semantic Knowledge for Word Sense Disambiguation

In Word Sense Disambiguation (WSD), supervised approaches are predominant in evaluation campaigns. The limited quantity of such corpora however restricts the coverage and the performance of these systems. In this article, we present two new methods that tackle this problem by exploiting the semantic relationships between senses such as synonymy, hypernymy and hyponymy, in order to compress the sense vocabulary of WordNet, and thus reduce the number of different sense tags that must be observed to disambiguate all words of the lexical database. Our methods greatly reduce the size of neural WSD models, with the benefit of improving their coverage without additional training data, and without impacting their precision. In addition to our methods, we present a neural WSD system which relies on the recent advances in contextualized word embeddings in order to achieve results that significantly outperform the state of the art on all WSD evaluation tasks.

MOTS-CLÉS : désambiguïsation lexicale, compression de vocabulaire, relations sémantiques.

KEYWORDS: Word Sense Disambiguation, Vocabulary Compression, Semantic Relationships.

1 Introduction

La Désambiguïsation Lexicale (DL) est une tâche qui vise à clarifier un texte en assignant à chacun de ses mots l'étiquette de sens la plus appropriée depuis un inventaire de sens prédéfini.

Il existe diverses approches pour la DL, telles que les approches à base de connaissances, qui

s'appuient sur des dictionnaires, des bases de données lexicales ou des graphes de connaissances couplés à des algorithmes tels que les mesures de similarité lexicale (Lesk, 1986) ou des mesures basées sur les graphes (Moro *et al.*, 2014), ou les méthodes supervisées, qui exploitent des corpus annotés en sens comme données d'apprentissage pour entraîner un classifieur multi-classe tel qu'un SVM (Chan *et al.*, 2007; Zhong & Ng, 2010), ou plus récemment un réseau neuronal (Kågebäck & Salomonsson, 2016). Les méthodes supervisées sont de loin les plus représentées car elles offrent généralement les meilleurs résultats dans les campagnes d'évaluation (par exemple (Navigli *et al.*, 2007)). Les classifieurs état de l'art combinaient jusqu'à récemment des caractéristiques précises telles que les parties du discours et les lemmes des mots voisins, (Zhong & Ng, 2010), mais ils sont maintenant remplacés par des réseaux de neurones récurrents qui apprennent leur propre représentation des mots (Raganato *et al.*, 2017; Le *et al.*, 2018; Vial *et al.*, 2019).

Cependant, une des limitations majeures des systèmes supervisés est la quantité limitée de corpus manuellement annotés en sens. En effet, le SemCor (Miller *et al.*, 1993), qui est le plus grand corpus manuellement annoté en sens disponible, contient 33 760 labels de sens différents, ce qui correspond à seulement environ 16% de l'inventaire de sens de WordNet [1] (Miller *et al.*, 1990), la base de données lexicale de référence largement utilisée en DL. De nombreux travaux tentent de résoudre ce problème via la création de nouveaux corpus annotés en sens, générés soit automatiquement (Pasini & Navigli, 2017), semi-automatiquement (Taghipour & Ng, 2015), ou bien par *crowdsourcing* (Yuan *et al.*, 2016), mais dans nos travaux, nous cherchons à résoudre ce problème en tirant parti des relations sémantiques présentes entre les sens de WordNet comme l'hyperonymie, l'hyponymie, l'antonymie, la méronymie, etc. Notre méthode est basée sur les observations suivantes :

1. Un sens et ses sens voisins dans le graphe des relations sémantiques de WordNet véhiculent tous une même idée ou concept, à des niveaux d'abstraction différents.
2. Dans certains cas, un mot peut être désambiguïsé en utilisant seulement les sens voisins de ses sens, et pas nécessairement ses sens propres.
3. Par conséquent, nous n'avons pas besoin de connaître tous les sens de WordNet pour désambiguïser tous les mots de WordNet.

Par exemple, considérons le mot « souris » et deux de ses sens : la souris *d'ordinateur* et la souris *l'animal*. Les notions plus générales comme « être vivant » (hyperonyme de souris/animal) et « appareil électronique » (hyperonyme de souris/ordinateur), permettent déjà de distinguer les deux sens, et toutes les notions plus spécialisées telles que « rongeur » ou « mammifère » sont, elles, superflues. En regroupant ces étiquettes de sens ensemble, on peut bénéficier de tous les autres exemples mentionnant un appareil électronique ou un être vivant dans un corpus d'entraînement, même si le mot « souris » n'est pas mentionné spécifiquement, pour désambiguïser le mot « souris ».

Contributions : Dans cet article, nous émettons l'hypothèse que seul un sous-ensemble des sens de WordNet peut être considéré pour pouvoir désambiguïser tous les mots de la base lexicale. Par conséquent, nous proposons deux méthodes différentes pour construire ce sous-ensemble que nous appelons méthodes de compression de vocabulaire de sens. En utilisant ces techniques, nous sommes en mesure d'améliorer considérablement la couverture des systèmes de DL supervisés, en éliminant quasiment le besoin d'une stratégie de repli habituellement employée pour les mots jamais observés pendant l'entraînement. Nous présentons des résultats qui surpassent l'état de l'art de façon significative sur toutes les tâches d'évaluation de la DL, et nous fournissons à la communauté notre outil ainsi que nos meilleurs modèles pré-entraînés, sur un dépôt GitHub dédié [2].

1. https://wordnet.princeton.edu/documentation/wnstats7wn
2. https://github.com/getalp/disambiguate

2 Désambiguïsation lexicale neuronale

Plusieurs avancées récentes ont été réalisées dans la création de nouvelles architectures neuronales pour les systèmes supervisés de désambiguïsation lexicale. Ces systèmes atteignent des performances état de l'art et certains peuvent intégrer des sources de connaissances externes. Dans cette section, nous donnons un aperçu de ces travaux.

2.1 Approches basées sur un modèle de langue

Dans ce type d'approches, initié par Yuan *et al.* (2016) et réimplémenté par Le *et al.* (2018), le composant principal est un modèle de langue neuronal capable de prédire un mot en tenant compte des mots qui l'entourent, grâce à un réseau neuronal entraîné sur une quantité massive de données non annotées (100 milliards de mots pour Yuan *et al.* (2016) et 1,8 milliards pour Le *et al.* (2018)).

Une fois le modèle de langue entraîné, il est utilisé pour produire des vecteurs de sens en moyennant les vecteurs de mots prédits par le modèle à l'endroit où ces mots sont annotés avec un sens particulier.

Au moment du test, le modèle de langue est utilisé pour prédire un vecteur en fonction du contexte environnant, et le sens le plus proche du vecteur prédit est attribué à chaque mot.

Ces systèmes ont l'avantage de contourner le problème de l'absence de données annotées en sens en concentrant le pouvoir d'abstraction offert par les réseaux neuronaux récurrents sur un modèle de langue de bonne qualité et entraîné de manière non supervisée. Cependant, ces méthodes souffrent toujours du manque de corpus annotés en sens étant donné qu'ils restent indispensables pour la création des vecteurs de sens.

2.2 Approches basées sur un classifieur linéaire et la fonction *softmax*

Dans ces systèmes, le réseau neuronal principal classifie et attribue directement un sens à chaque mot donné en entrée à l'aide d'une distribution de probabilité calculée par la fonction *softmax*. Les annotations en sens sont simplement considérées comme des balises placées sur chaque mot, à la manière d'une tâche d'étiquetage en parties du discours par exemple.

On peut distinguer deux branches distinctes de ces types de réseaux neuronaux :

1. Ceux dans lesquels il y a un réseau neuronal (ou classifieur) distinct et spécifique à chaque lemme du dictionnaire (Iacobacci *et al.*, 2016; Kågebäck & Salomonsson, 2016). Chaque classifieur est capable de gérer un lemme particulier avec ses sens. Par exemple, l'un des classifieurs est spécialisé dans le choix entre les quatre sens possibles du nom « souris ». Ce type d'approche est particulièrement adapté aux tâches de *lexical sample*, où seul un petit nombre de mots distincts et très ambigus doivent être annotés dans plusieurs contextes. Mais ils nécessiteraient plusieurs milliers de réseaux différents [3] pour pouvoir aussi être utilisés dans les tâches de désambiguïsation lexicales *all words*, dans lesquelles tous les mots d'un document doivent être annotés en sens.
2. Ceux dans lesquels il y a un seul réseau neuronal, plus grand et capable de gérer tous les lemmes du lexique, qui attribuent à un mot un sens issu de l'ensemble de tous les sens de l'inventaire de sens utilisé (Raganato *et al.*, 2017; Vial *et al.*, 2019).

L'avantage de la première branche est que pour désambiguïser un mot, il est beaucoup plus facile de

3. L'ensemble de WordNet contient par exemple 26 896 mots polysémiques (`https://wordnet.princeton.edu/documentation/wnstats7wn`)

limiter notre choix à l'un de ses sens possibles que de chercher parmi tous les sens de tous les mots du lexique. Pour se donner une idée, le nombre moyen de sens des mots polysémiques dans WordNet est d'environ 3, alors que le nombre total de sens en considérant tous les mots est 206 941. [4]

La seconde approche a cependant une propriété intéressante : tous les sens résident dans le même espace vectoriel et partagent donc des caractéristiques dans les couches cachées du réseau. Cela permet au modèle de prédire un sens identique pour deux mots différents (synonymes), mais aussi de prédire un sens pour un mot non présent dans le dictionnaire (néologisme, faute d'orthographe, etc.).

Enfin, dans deux articles récents, Luo *et al.* (2018a,b) ont proposés une amélioration de ce type d'architectures, en calculant une attention entre le contexte d'un mot cible et les définitions de ses différents sens. Ainsi, leur travail est le premier à incorporer les connaissances de WordNet dans un système de désambiguïsation neuronal.

3 Compression de vocabulaire de sens

Les systèmes supervisés neuronaux état de l'art tels que Yuan *et al.* (2016); Raganato *et al.* (2017); Le *et al.* (2018); Luo *et al.* (2018a,b); Vial *et al.* (2019) sont tous confrontés aux mêmes limitations :

1. La quantité de données annotés manuellement en sens étant très limitée, il se peut qu'un mot cible ne soit jamais observé pendant l'entraînement. Dans ce cas, le système ne peut pas être en mesure de l'annoter, et une stratégie de repli est généralement effectuée (par exemple utiliser le premier sens du mot dans WordNet).
2. Pour la même raison, un mot peut être observé, mais pas tous ses sens. Dans ce cas, le système va être capable d'annoter ce mot, mais si le sens attendu n'a jamais été observé, le résultat sera faux, quelle que soit l'architecture sous-jacente du système supervisé.
3. L'empreinte mémoire des modèles neuronaux ainsi que leur temps d'entraînement et d'exécution augmentent avec la quantité de données d'apprentissage et le nombre d'étiquettes de sens différentes prises en compte, nombre qui monte jusqu'à 206 941 si l'on considère toutes les étiquettes de sens de WordNet.

Afin de résoudre ces problèmes, nous proposons deux nouvelles méthodes permettant de regrouper ensemble des étiquettes de sens qui se réfèrent à des concepts similaires, tout en nous assurant que ces groupes de sens permettent toujours de discriminer les différents sens de tous les mots du lexique, afin de retrouver l'étiquette de sens originale pour un mot au moment de le désambiguïser. En conséquence, le vocabulaire de sens, c'est à dire le nombre total d'étiquettes de sens dans notre inventaire de sens diminue, le système est capable de mieux généraliser, et sa couverture augmente.

3.1 Des sens aux *synsets* : une première compression de vocabulaire de sens à travers la synonymie

Dans la base de données lexicale WordNet (Miller *et al.*, 1990), les sens sont organisés en ensembles de synonymes appelés *synsets*. Un *synset* est concrètement un groupe d'un ou plusieurs sens qui ont la même définition et donc la même signification. Par exemple, les premiers sens des mots « *eye* », « *optic* » et « *oculus* » appartiennent tous au même *synset* dont la définition est « l'organe de la vue ».

La conversion des étiquettes de sens (« Xème sens du mot N ») aux étiquettes de *synsets* (« *synset* numéro Y »), illustré dans la figure 1, est ainsi une façon de compresser le vocabulaire qui est

4. https://wordnet.princeton.edu/documentation/wnstats7wn

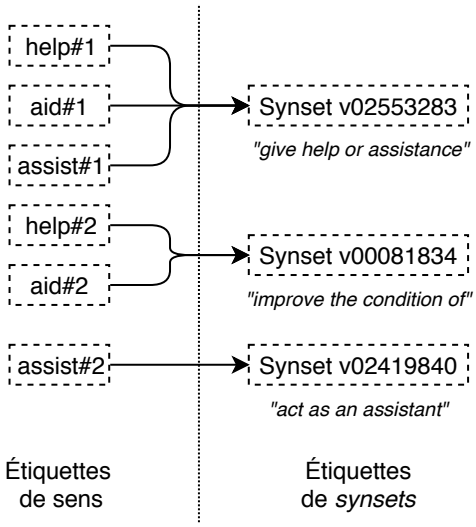

FIGURE 1 – Conversion des étiquettes de sens vers des étiquettes de *synsets*, appliqué aux deux premiers sens des mots « *help* », « *aid* » et « *assist* ». Le nombre de sens différents dans notre vocabulaire passe ainsi de six à trois.

déjà appliquée dans plusieurs travaux (Yuan *et al.*, 2016; Le *et al.*, 2018; Vial *et al.*, 2019) sans être toujours explicitement précisée. Cette méthode contribue pourtant clairement à améliorer la couverture des systèmes supervisés. En effet, si le verbe « *aid* » annoté avec son premier sens est observé dans les données d'apprentissage, le contexte autour du mot cible peut être aussi utile pour annoter ultérieurement les verbes « *assist* » ou « *help* » avec la même étiquette de *synset*.

En allant plus loin, on peut trouver d'autres informations dans WordNet qui peuvent aider à mieux généraliser. La première nouvelle méthode que nous proposons repose ainsi sur ce même principe de regroupement de sens, mais en exploitant les relations d'hyperonymie et d'hyponymie entre les sens.

3.2 Compression de vocabulaire de sens à travers les relations d'hyperonymie, d'hyponymie et d'instance

Selon Polguère (2003), l'hyperonymie et l'hyponymie sont deux relations sémantiques qui correspondent à un cas particulier d'inclusion de sens : l'hyponyme d'un terme est une spécialisation de ce terme, alors que son hyperonyme est une généralisation. Par exemple, une « souris » est un type de « rongeur » qui est à son tour un type d'« animal ». Dans WordNet, ces relations lient presque tous les noms ensemble allant de la racine générique, le nœud « entité » aux feuilles les plus spécifiques, par exemple « souris à pattes blanches ». Si l'on prend aussi en compte la relation d'instance, qui fonctionne de la même manière mais qui lie les entitées nommées aux noms courants (par exemple « Einstein » est une instance de « physicien »), tous les noms de WordNet font partie de cette même hiérarchie.

Ces relations sont également présentes sur plusieurs verbes : ainsi, par exemple, « additionner » est une manière de « calculer » qui est à son tour une manière de « raisonner ».

Pour la DL, tout comme le regroupement des synonymes en *synsets* aide à mieux généraliser, nous faisons l'hypothèse que le regroupement des sens faisant partie d'une même hiérarchie d'hyperonymie va aussi aider à mieux généraliser, et que les concepts les plus spécialisés de WordNet sont souvent superflus. En effet, si l'on considère un sous-ensemble de WordNet qui ne comprend que le mot « souris », avec son premier sens (le petit rongeur), son quatrième sens (le dispositif électronique), et tous leurs hyperonymes, tel qu'illustré dans la figure 2, on voit que les concepts « artefact » et « être vivant » suffisent à différencier les deux sens, et toutes les étiquettes plus spécialisées pourrait être ramenés à ces deux concepts. Ainsi, non seulement le vocabulaire de sens, c'est à dire le nombre

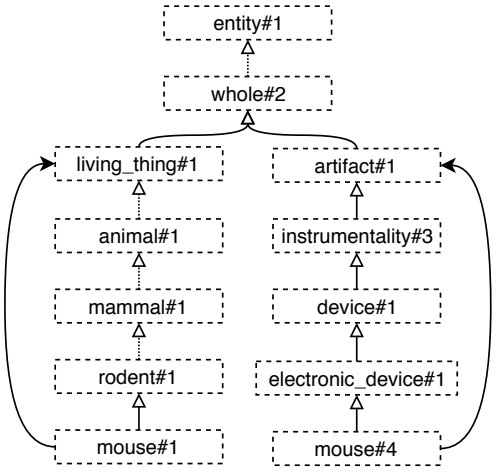

FIGURE 2 – Compression de vocabulaire utilisant la hiérarchie d'hyperonymie, appliquée au premier et quatrième sens du mot « *mouse* ». Les lignes en pointillés indiquent que des nœuds ont été omis pour la clarté.

d'étiquettes de sens dans notre inventaire, sera réduit, mais en plus tous les autres « êtres vivants » donneront des exemples qui pourront ensuite permettre de différencier les deux sens de souris.

En considérant maintenant tout le vocabulaire de WordNet, l'objectif de notre méthode est ainsi de faire correspondre chaque sens à son ancêtre le plus haut dans sa hiérarchie d'hyperonymie, avec les contraintes suivantes : Premièrement, cet ancêtre doit permettre de discriminer tous les différents sens du mot cible. Deuxièmement, nous devons conserver les hyperonymes qui sont indispensables pour discriminer les sens des autres mots du dictionnaire. Par exemple, en prenant tout WordNet en considération, nous ne pouvons pas faire correspondre « souris#1 » à « être vivant#1 », parce qu'une étiquette plus spécifique, « animal#1 » est nécessaire pour distinguer les deux sens du mot « proie » (un sens décrit une personne et l'autre un animal).

Notre méthode fonctionne donc en deux étapes :

1. Nous marquons comme « nécessaires » les enfants du premier ancêtre commun de chaque paire de sens de chaque mot de WordNet.
2. Nous faisons correspondre chaque sens à son ancêtre le plus bas dans sa hiérarchie d'hyperonymie ayant été précédemment marqué comme « nécessaire ».

En conséquence, les sens les plus spécifiques de l'arbre qui ne sont pas indispensables pour distinguer un mot de l'inventaire lexical seront automatiquement supprimés du vocabulaire. En d'autres termes, l'ensemble de sens qui reste dans le vocabulaire est le plus petit sous-ensemble de tous les *synsets* qui sont nécessaires pour distinguer chaque sens de chaque mot de WordNet, en considérant seulement les liens d'hyperonymie, d'hyponymie et d'instance.

3.3 Compression de vocabulaire de sens à travers l'ensemble des relations sémantiques de WordNet

En plus de l'hyperonymie, de l'hyponymie et de la relation d'instance, WordNet contient plusieurs autres relations entre *synsets*, telles que la méronymie (X fait partie de Y, ou X est un membre de Y) et son opposé l'holonymie, l'antonymie (X est le contraire de Y) et son opposé la similarité, etc.

Nous proposons ainsi une deuxième nouvelle méthode de compression du vocabulaire de sens, qui prend en compte toutes les relations sémantiques offertes par WordNet, afin de former des groupes de *synsets* proches.

Par exemple, en utilisant toutes les relations sémantiques disponibles, nous pourrions former un groupe contenant « physicien », « physique » (domaine), « Einstein » (instance), « astronome » (hyponyme), mais aussi d'autres sens connexes tels que « photon », car c'est un méronyme de « rayonnement », qui est un hyponyme de « énergie », qui appartient au même domaine de « physique », etc.

Notre méthode fonctionne en construisant ces groupes de manière itérative. Soit S l'ensemble des *synsets* de WordNet et C l'ensemble des groupes de *synsets* que l'on cherche à construire, on initialise d'abord C comme des singletons contenant chacun un *synset* différent.

$$S = \{s_0, s_1, ..., s_n\} \qquad C = \{c_0, c_1, ..., c_n\} \qquad C = \{\{s_0\}, \{s_1\}, ..., \{s_n\}\}$$

Ensuite, à chaque étape, on trie C par taille de groupes, et on sélectionne le plus petit groupe c_x ainsi que le plus petit groupe relié à c_x, c_y. On considère qu'un groupe c_a est relié à un groupe c_b si un *synset* $s_a \in c_a$ est relié à un *synset* $s_b \in c_b$ par n'importe quel lien sémantique. On fusionne c_x et c_y ensemble, si et seulement si l'opération permet toujours de discriminer les différents sens de tous les mots de la base lexicale. Si c'est le cas, on valide la fusion et on passe à l'étape suivante. Si ce n'est pas le cas, on annule la fusion et on essaye avec un autre groupe relié à c_x. S'il est impossible de fusionner un groupe avec c_x, alors on essaye avec le plus petit groupe suivant, et si aucune fusion n'est possible pour aucun des groupes, l'algorithme s'arrête.

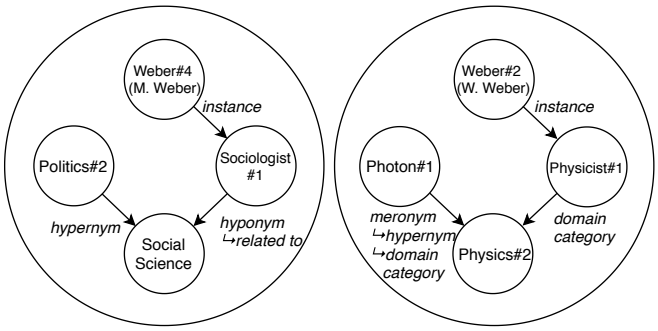

FIGURE 3 – Exemple de groupes de sens pouvant résulter de notre méthode, si on ne considère que deux sens du nom « Weber » et seulement certaines relations.

Dans la figure 3, nous montrons un ensemble possible de groupes qui pourraient résulter de notre méthode.

Cette méthode produit des groupes significativement plus grands que celle s'appuyant sur les hyperonymes. En effet, en moyenne, un groupe contient 5 *synsets* avec cette dernière, alors qu'il en contient 17 avec cette méthode. De plus, cette méthode, contrairement à la précédente, est également stochastique, parce qu'à chaque fois qu'on ordonne les groupes par taille, l'algorithme de tri place les groupes de même taille dans un ordre aléatoire. Cependant, comme nous réordonnons les groupes après chaque fusion, les groupes sont de taille assez équilibrés, et nous avons observé que la taille finale du vocabulaire (c.-à-d. le nombre de groupes) se situe toujours entre 11 000 et 13 000,

Dans la suite, on considère un ensemble C généré après que l'algorithme se soit arrêté après 105 775 étapes de fusion (générant ainsi 11 885 groupes de sens).

La table 1 montre l'effet de la compression de vocabulaire via les synonymes (sens vers *synsets*), de notre première nouvelle méthode utilisant les hyperonymes, ainsi que de notre deuxième nouvelle méthode utilisant toutes les relations de WordNet, sur la taille du vocabulaire de sens de WordNet, et sur la couverture du SemCor. Comme nous pouvons le constater, la taille du vocabulaire diminue considérablement grâce à nos méthodes, et la couverture d'un même corpus est nettement améliorée.

Méthode de compression	Taille du vocabulaire	Taux de compression	Couverture du SemCor
Référence	206 941	référence	16%
Synonymes	117 659	43%	22%
Hyperonymes	39 147	81%	32%
Toutes relations	11 885	94%	39%

TABLE 1 – Résultats de nos deux méthodes de compression de vocabulaire sur la taille du vocabulaire et la couverture du SemCor. La couverture correspond au ratio du nombre d'étiquettes de sens différentes observables dans le corpus sur le nombre total d'étiquettes (taille du vocabulaire).

4 Protocole expérimental

Afin d'évaluer nos méthodes de compression de vocabulaire de sens, nous les avons appliquées à un système neuronal de DL état de l'art similaire à celui de Vial *et al.* (2019) (voir la section 2.2). Notre réseau de neurones prend ainsi en entrée directement les mots sous forme vectorielle, à partir d'un modèle de vecteurs de mots pré-entraîné, il repose ensuite sur une ou plusieurs couches cachées, puis sur une couche de sortie, qui associe à chaque mot une distribution de probabilité sur tous les sens du vocabulaire utilisé, à l'aide de la fonction *softmax*.

4.1 Détails de l'architecture

En entrée de notre réseau, nous avons utilisé les vecteurs contextualisés BERT (Devlin *et al.*, 2018). Nous avons utilisé le modèle pour l'anglais « bert-large-cased » qui est pré-entraîné sur BookCorpus (Zhu *et al.*, 2015) et Wikipedia, et qui produit des vecteurs de dimension 1 024.

Pour les couches cachées, nous avons appliqué 6 couches d'encodeurs *Transformer* (Vaswani *et al.*, 2017), avec les même paramètres que le modèle « base » de l'article original (8 têtes d'attention, dimension cachée de 2 048, et régularisation *dropout* à 0,1). Les couches *Transformer* sont basés sur le mécanisme d'auto-attention, et nous les avons utilisés à la place des cellules récurrentes plus classiques comme des *LSTM* ou des *GRU*, parce que plusieurs travaux récents ont montré leur plus grande efficacité dans une multitude de tâches, par exemple en traduction automatique (Vaswani *et al.*, 2017; Ott *et al.*, 2018) et en modélisation de la langue (Devlin *et al.*, 2018).

De plus, étant donné que les vecteurs retournés par BERT encodent directement les positions des mots, il n'est pas nécessaire d'avoir une récurrence au niveau des couches cachées. Ainsi, nous n'ajoutons pas de vecteurs de positions supplémentaires en entrée de notre encodeur.

Pour tous les autres paramètres du modèle, comme le nombre de phrase par mini-lot, et la méthode d'optimisation, nous avons utilisé les mêmes paramètres que Vial *et al.* (2019).

4.2 Entraînement du modèle

Nous avons comparé nos méthodes sur deux ensembles de corpus d'entraînement : le SemCor (Miller *et al.*, 1993), le plus grand corpus annoté en sens utilisé pour l'apprentissage de la plupart des systèmes supervisés de DL, et la concaténation du SemCor et du WordNet Gloss Tagged [5]. Ce dernier est un corpus distribué dans WordNet depuis sa version 3.0, et il contient les définitions de tous les sens de WordNet, annoté manuellement ou semi-automatiquement en sens. Nous avons utilisé les versions de

5. http://wordnetcode.princeton.edu/glosstag-files/glosstag.shtml

ces corpus fournies avec la ressource UFSAC 2.1 [6] (Vial *et al.*, 2018).

Système	SE2	SE3	SE07 17	SE13	SE15	ALL	SE07 07
SemCor, référence	91,15	96,76	97,58	91,06	94,78	93,23	92,84
SemCor, hyperonymes	98,03	99,19	99,78	99,15	98,39	98,75	98,85
SemCor, toutes relations	99,56	99,84	100	100	98,92	99,67	99,69
SemCor+WNGT, référence	97,81	98,92	99,34	97,63	99,34	98,26	98,45
SemCor+WNGT, hyperonymes	99,74	99,95	100	99,76	99,91	99,83	99,91
SemCor+WNGT, toutes relations	100	100	100	100	99,91	99,99	100
Nombre de mots à annoter	2282	1850	455	1644	1022	7253	2261

TABLE 2 – Couverture (en %) des corpus d'évaluation en fonction du corpus d'apprentissage et de l'utilisation de notre méthode de compression de vocabulaire.

Nous avons choisi d'ajouter uniquement le WNGT en plus du SemCor à nos données d'entraînement, et pas tous les corpus de la ressource UFSAC 2.1, parce que c'est le seul, avec le SemCor, dans lequel à la fois tous les mots sont annotés en sens, l'inventaire de sens utilisé par les annotateurs est directement WordNet, les annotations ne sont pas entièrement automatiques, et la ressource est libre. Nous avons ainsi cherché à utiliser seulement des données de la meilleure qualité possible, pour éviter d'ajouter du bruit et/ou de rallonger le temps d'entraînement de nos modèles.

Nous avons entraîné chaque modèle sur 20 passes de nos données d'entraînement. Au début de chaque passe, nous avons mélangé toutes les phrases aléatoirement, et à la fin de chaque passe, nous avons évalué notre modèle sur un jeu de développement, et nous avons conservé celui qui a obtenu le meilleur score F1 de DL. Le corpus de développement est constitué de 4 000 phrases prises aléatoirement du WNGT pour le système entraîné sur le SemCor seul, et de 4 000 phrases extraites aléatoirement de nos données d'entraînement pour les autres.

Nous avons ainsi entraînés trois systèmes :

1. un système « référence » dont le vocabulaire de sens est celui de tous les *synsets* vus pendant l'entraînement (utilisant ainsi la compression classique via les synonymes) ;
2. un système « hyperonymes » entraîné dans les mêmes conditions, mais avec notre première méthode de compression du vocabulaire via les hyperonymes, les hyponymes et les instances appliquée sur le corpus d'entraînement ;
3. un système « toutes relations » qui applique cette fois-ci sur le corpus d'entraînement notre deuxième méthode de compression de vocabulaire via toutes les relations sémantiques de WordNet.

	Nombre de paramètres	
Système	SemCor	SemCor+WNGT
Référence	77,15M	120,85M
Hyperonymes	63,44M	79,85M
Toutes relations	55,16M	60,27M

TABLE 3 – Nombre de paramètres d'un modèle en fonction du corpus d'apprentissage et de notre méthode de compression de vocabulaire.

Tous les entraînements ont été effectués sur un seul GPU Titan X de Nvidia. Dans la table 3, nous montrons le nombre de paramètres des différents modèles, en fonction du corpus d'entraînement et de notre méthode de compression du vocabulaire. Comme nous pouvons le voir, ce nombre est réduit par un facteur de 1,2 à 2 grâce à nos méthodes de compression.

6. https://github.com/getalp/UFSAC

4.3 Résultats

Nous avons évalué nos modèles sur tous les corpus d'évaluation de la DL de l'anglais des campagnes d'évaluation SensEval/SemEval, c'est-à-dire les corpus d'évaluation « grain fin » de SensEval 2 (Edmonds & Cotton, 2001), SensEval 3 (Snyder & Palmer, 2004), SemEval 2007 (tâche 17) (Pradhan *et al.*, 2007), SemEval 2013 (Navigli *et al.*, 2013) et SemEval 2015 (Moro & Navigli, 2015), ainsi que le corpus « ALL » constitué de leur concaténation. Nous avons également comparé nos résultats sur la tâche « gros grain » de SemEval 2007 (tâche 7) (Navigli *et al.*, 2007).

Pour chaque évaluation, nous avons entraîné 8 modèles indépendant, et nous donnons le score obtenu par un système « ensemble » qui moyenne leurs prédictions à l'aide d'une moyenne géométrique.

Les scores obtenus par nos systèmes en comparaison avec les meilleurs systèmes de l'état de l'art et l'étalon du premier sens sont présents dans le tableau 4, et le tableau 2 montre la couverture de nos systèmes sur les tâches d'évaluation.

	SE2	SE3	SE07	SE13	SE15	ALL (concat. tâches précédentes)					SE07
Système			17			noms	verbes	adj.	adv.	total	07
Étalon du premier sens	65,6	66,0	54,5	63,8	67,1	67,7	49,8	73,1	80,5	65,5	78,9
UFSAC+1M (Vial *et al.*, 2019)	74,6	69,4	60,7	69,8	74,2	-	-	-	-	†71,1	85,0
HCAN (Luo *et al.*, 2018a)	72,8	70,3	-	68,5	72,8	72,7	58,2	77,4	84,1	71,1	-
LSTMLP (Yuan *et al.*, 2016)	73,8	71,8	63,5	69,5	72,6	†73,9	-	-	-	†71,5	83,6
SemCor, référence	77,2	76,5	70,1	74,7	77,4	78,7	65,2	79,1	85,5	76,0	87,7
SemCor, hyperonymes	77,5	77,4	69,5	76,0	78,3	79,6	65,9	79,5	85,5	76,7	87,6
SemCor, toutes relations	76,6	76,9	69,0	73,8	75,4	77,2	66,0	80,1	85,0	75,4	86,7
SemCor+WNGT, référence	**79,7**	76,1	**74,1**	78,6	80,4	80,6	68,1	82,4	86,1	78,3	90,4
SemCor+WNGT, hyperonymes	**79,7**	77,8	73,4	**78,7**	**82,6**	81,4	68,7	83,7	85,5	**79,0**	90,4
SemCor+WNGT, toutes relations	79,4	**78,1**	71,4	77,8	81,4	80,7	68,6	82,8	85,5	78,5	**90,6**

TABLE 4 – Scores F1 (%) sur les tâches de DL de l'anglais des campagnes d'évaluation SensEval/SemEval. La tâche « ALL » est la concaténation de SE2, SE3, SE07 17, SE13 et SE15. La stratégie de repli est appliquée sur les mots dont aucun sens n'a été observé pendant l'entraînement. Les scores en **gras** sont à notre connaissance les meilleurs résultats obtenus sur la tâche. Les scores prefixés par une obélisque (†) ne sont pas fournis par les auteurs mais sont déduits de leurs autres scores.

Concernant les résultats présentés dans la table 4, nous observons que nos systèmes qui utilisent nos méthodes de compression de vocabulaire, que ce soit à travers les relations d'hyperonymie ou à travers toutes les relations obtiennent des scores qui sont globalement équivalents ou légèrement supérieurs aux systèmes « référence » qui n'utilisent pas nos méthodes.

Nos méthodes de compression améliorent cependant grandement la couverture de nos systèmes. En effet, comme nous pouvons le voir dans la table 2, sur un total de 7 253 mots à annoter pour le corpus « ALL », le système de référence entraîné sur le SemCor n'est pas capable d'annoter 491 d'entre eux, alors qu'avec la compression du vocabulaire à travers les hyperonymes, ce nombre descend à 91, et 24 avec la compression à travers toutes les relations.

Lors de l'ajout du WordNet Gloss Tagged aux données d'entraînement, seulement 12 mots ne peuvent pas être annotés avec le système « hyperonymes », et avec le système « toutes relations », plus qu'un

Corpus d'entraînement	Vecteurs de mots pré-entraînés	Ensemble	Scores F1 sur la tâche "ALL" (%)					
			Référence		Hyperonymes		Toutes relations	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
SemCor+WNGT	BERT	Oui	78,27	-	79,00	-	78,48	-
SemCor+WNGT	BERT	Non	76,97	±0,38	77,08	±0,17	76,52	±0,36
SemCor+WNGT	ELMo	Oui	75,16	-	74,65	-	70,58	-
SemCor+WNGT	ELMo	Non	74,56	±0,27	74.36	±0,27	68,77	±0.30
SemCor+WNGT	GloVe	Oui	72,23	-	72,74	-	71,42	-
SemCor+WNGT	GloVe	Non	71,93	±0,35	71,79	±0,29	69,60	±0,32
SemCor	BERT	Oui	76,02	-	76,73	-	75,40	-
SemCor	BERT	Non	75,06	±0,26	75,59	±0,16	73,91	±0,33
SemCor	ELMo	Oui	72,55	-	73,09	-	69,43	-
SemCor	ELMo	Non	72,21	±0,13	72,83	±0,24	68,74	±0,29
SemCor	GloVe	Oui	70,77	-	71,18	-	68,44	-
SemCor	GloVe	Non	70,51	±0,16	70,77	±0,21	67,48	±0,55
Système « élève » (Vial *et al.*, 2019)								
SemCor+UFSAC+1M News 2016	GloVe	Oui	71,1					
HCAN (Luo *et al.*, 2018a)								
SemCor+WordNet glosses	GloVe	Non	71,1					
LSTMLP (Yuan *et al.*, 2016)								
SemCor+1K (private)	private	Non	71,5					

TABLE 5 – Étude des hyperparamètres sur la tâche "ALL" (concaténation des corpus de toutes les tâches de désambiguïsation lexicale à granularité fine de SensEval/SemEval). Pour les systèmes qui n'utilisent pas l'ensemble, nous montrons la moyenne des scores ($\bar{x}$) de huit modèles entraînés séparément, avec l'écart type (σ).

seul mot (l'adjectif monosémique « cytotoxic ») ne peut pas être annoté parce que son sens n'a pas été vu pendant l'entraînement. Si nous prenons en compte uniquement les mots polysémiques, le système basé sur la compression à travers toutes les relations et entraîné sur le SemCor n'est pas capable d'annoter seulement un seul mot (l'adverbe « eloquently »). Avec le WNGT en plus, il a une couverture de 100%.

Par rapport aux autres travaux, nous obtenons des résultats surpassant significativement l'état de l'art dans toutes les tâches, notamment grâce à l'ajout du WordNet Gloss Tagged aux données d'entraînement, et des vecteurs BERT en entrée de notre système.

4.4 Étude des hyperparamètres

Afin de mieux comprendre l'origine de nos scores, nous étudions l'impact de nos principaux paramètres sur les résultats. En plus du corpus d'entraînement et de la méthode de compression du vocabulaire, nous avons choisi deux paramètres qui nous différencient de l'état de l'art : le modèle de vecteurs de mots pré-entraînés, et la méthode d'ensemble, et nous les avons fait varier.

Pour le modèle de vecteurs de mots, nous avons expérimenté avec BERT (Devlin *et al.*, 2018) comme pour nos résultats principaux, mais aussi avec ELMo (Peters *et al.*, 2018) et GloVe (Pennington *et al.*, 2014). Pour ELMo, nous avons utilisé le modèle entraîné sur Wikipedia et les données monolingues de WMT 2008-2012. [7] Pour GloVe, nous avons utilisé le même modèle que Luo *et al.* (2018a) et Vial

7. https://allennlp.org/elmo

et al. (2019) entraîné sur Wikipedia 2014 et Gigaword 5. [8] Comme les représentations vectorielles de GloVe n'encodent pas la position des mots (un mot a la même représentation quelque soit sa position ou son contexte), nous avons réutilisé une couche de cellules *LSTM* bidirectionnelles de taille 1 000 par direction pour les couches cachées (comme Vial *et al.* (2019)).

Pour la méthode d'ensemble, nous avons expérimenté soit en l'utilisant, comme dans nos résultats principaux, c'est à dire en moyennant les prédictions de 8 modèles entraînés séparément, ou bien en donnant la moyenne et l'écart type des scores des 8 modèles évalués individuellement.

Comme nous pouvons le voir dans la table 5, le corpus d'entraînement supplémentaire (WNGT) et encore plus l'utilisation de BERT en tant que vecteurs de mots ont tous les deux un impact majeur sur nos résultats et conduisent à des scores supérieurs à l'état de l'art. L'utilisation de BERT au lieu de ELMo ou GloVe améliore respectivement le score d'environ 3 et 5 points dans chaque expérience, et l'ajout du WNGT aux données de d'entraînement l'améliore encore d'environ 2 points. Enfin, l'utilisation d'ensembles ajoute environ 1 point au score F1 final.

Enfin, à travers les scores obtenus par les modèles individuels (sans ensemble), nous pouvons observer sur les écarts-types que la méthode de compression du vocabulaire par les hyperonymes n'a jamais d'impact significatif sur le score final. Cependant, la méthode de compression via toutes les relations semble avoir un impact négatif sur les résultats dans certains cas (en utilisant GloVe et ELMo particulièrement, et en utilisant le SemCor seul comme corpus d'entraînement).

5 Conclusion

Dans cet article, nous avons présenté deux nouvelles méthodes qui améliorent la couverture et la capacité de généralisation des systèmes de DL supervisés, en réduisant le nombre d'étiquettes de sens différentes dans WordNet afin de ne conserver que celles qui sont essentielles pour différencier les sens de tous les mots présents dans la base lexicale. À l'échelle de l'ensemble de la base de données lexicale, nous avons montré que ces méthodes permettaient de réduire le nombre total d'étiquettes de sens différentes dans WordNet à seulement 6% de sa taille originale, et que la couverture d'un même corpus d'entraînement est ensuite plus que doublée.

Nous avons entraîné un système de DL neuronal état de l'art et nous avons montré que nos méthodes permettaient de réduire la taille des modèles par un facteur de 1,2 à 2 et de largement augmenter leur couverture, sans dégrader leurs performances. Au final, nous obtenons une couverture de 99,99% sur l'ensemble des tâches d'évaluation (soit un seul mot manquant sur les 7 253) lorsque l'on entraîne notre système sur le SemCor uniquement, et 100% lorsque l'on ajoute le WordNet Gloss Tagged aux données d'entraînement. On élimine ainsi quasiment le besoin d'une méthode de repli pour désambiguïser n'importe quel mot du vocabulaire de WordNet.

Notre méthode combinée avec les récentes avancées en terme de vecteurs de mots pré-entraînés permet à notre système de surpasser nettement l'état de l'art dans toutes les tâches d'évaluation de la DL de l'anglais, avec une bien meilleure couverture.

Pour finir, bien que nous ayons appliqué nos méthodes uniquement sur l'anglais dans cet article, elles peuvent facilement s'appliquer à d'autres langues en utilisant toujours WordNet comme inventaire de sens. Par exemple, elles peuvent s'appliquer à la désambiguïsation d'une langue moins bien dotée en utilisant la méthode de Hadj Salah *et al.* (2018). Cependant, du fait que les autres langues ont très peu de ressources annotées manuellement en sens (que ce soit avec l'inventaire de sens WordNet ou un autre), l'évaluation des systèmes de DL pour d'autres langues que l'anglais est limité.

8. https://nlp.stanford.edu/projects/glove/

Références

CHAN Y. S., NG H. T. & ZHONG Z. (2007). Nus-pt : Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, p. 253–256, Stroudsburg, PA, USA : Association for Computational Linguistics.

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

EDMONDS P. & COTTON S. (2001). Senseval-2 : Overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, SENSEVAL '01, p. 1–5, Stroudsburg, PA, USA : Association for Computational Linguistics.

HADJ SALAH M., VIAL L., BLANCHON H., ZRIGUI M., LECOUTEUX B. & SCHWAB D. (2018). Traduction automatique de corpus en anglais annotés en sens pour la désambiguïsation lexicale d'une langue moins bien dotée, l'exemple de l'arabe. In *25e conférence sur le Traitement Automatique des Langues Naturelles*, Rennes, France.

IACOBACCI I., PILEHVAR M. T. & NAVIGLI R. (2016). Embeddings for word sense disambiguation : An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 897–907, Berlin, Germany : Association for Computational Linguistics.

KÅGEBÄCK M. & SALOMONSSON H. (2016). Word sense disambiguation using a bidirectional lstm. In *5th Workshop on Cognitive Aspects of the Lexicon (CogALex)* : Association for Computational Linguistics.

LE M., POSTMA M., URBANI J. & VOSSEN P. (2018). A deep dive into word sense disambiguation with lstm. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 354–365 : Association for Computational Linguistics.

LESK M. (1986). Automatic sense disambiguation using mrd : how to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC '86*, p. 24–26, New York, NY, USA : ACM.

LUO F., LIU T., HE Z., XIA Q., SUI Z. & CHANG B. (2018a). Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 1402–1411 : Association for Computational Linguistics.

LUO F., LIU T., XIA Q., CHANG B. & SUI Z. (2018b). Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2473–2482 : Association for Computational Linguistics.

MILLER G. A., BECKWITH R., FELLBAUM C., GROSS D. & MILLER K. (1990). Wordnet : An on-line lexical database. *International Journal of Lexicography*, **3**, 235–244.

MILLER G. A., LEACOCK C., TENGI R. & BUNKER R. T. (1993). A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, p. 303–308, Stroudsburg, PA, USA : Association for Computational Linguistics.

MORO A. & NAVIGLI R. (2015). Semeval-2015 task 13 : Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, p. 288–297, Denver, Colorado : Association for Computational Linguistics.

MORO A., RAGANATO A. & NAVIGLI R. (2014). Entity linking meets word sense disambiguation : a unified approach. *TACL*, **2**, 231–244.

NAVIGLI R., JURGENS D. & VANNELLA D. (2013). SemEval-2013 Task 12 : Multilingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, p. 222–231.

NAVIGLI R., LITKOWSKI K. C. & HARGRAVES O. (2007). Semeval-2007 task 07 : Coarse-grained english all-words task. In *SemEval-2007*, p. 30–35, Prague, Czech Republic.

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

PASINI T. & NAVIGLI R. (2017). Train-o-matic : Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 78–88 : Association for Computational Linguistics.

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. In *Proc. of NAACL*.

POLGUÈRE A. (2003). *Lexicologie et sémantique lexicale*. Les Presses de l'Université de Montréal.

PRADHAN S. S., LOPER E., DLIGACH D. & PALMER M. (2007). Semeval-2007 task 17 : English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, p. 87–92, Stroudsburg, PA, USA : Association for Computational Linguistics.

RAGANATO A., DELLI BOVI C. & NAVIGLI R. (2017). Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 1167–1178 : Association for Computational Linguistics.

SNYDER B. & PALMER M. (2004). The english all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

TAGHIPOUR K. & NG H. T. (2015). One Million Sense-Tagged Instances for Word Sense Disambiguation and Induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, p. 338–344, Beijing, China : Association for Computational Linguistics.

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

VIAL L., LECOUTEUX B. & SCHWAB D. (2018). UFSAC : Unification of Sense Annotated Corpora and Tools. In *Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

VIAL L., LECOUTEUX B. & SCHWAB D. (2019). Approche supervisée à base de cellules LSTM bidirectionnelles pour la désambiguïsation lexicale. *Traitement Automatique des Langues*.

YUAN D., RICHARDSON J., DOHERTY R., EVANS C. & ALTENDORF E. (2016). Semi-supervised word sense disambiguation with neural models. In *COLING 2016*.

ZHONG Z. & NG H. T. (2010). It makes sense : A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, p. 78–83, Stroudsburg, PA, USA : Association for Computational Linguistics.

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. *2015 IEEE International Conference on Computer Vision (ICCV)*.

Corpus annoté de cas cliniques en français

Natalia Grabar[1] Cyril Grouin[2] Thierry Hamon[2, 3] Vincent Claveau[4]
(1) CNRS, UMR 8163 ; Univ. Lille, UMR 8163 - STL - Savoirs Textes Langage, F-59000, Lille, France
(2) LIMSI, CNRS, Université Paris-Saclay, F-91405 Orsay, France
(3) Université Paris 13, Sorbonne Paris Cité, F-93430 Villetaneuse, France
(4) Univ Rennes, Inria, CNRS, IRISA, F-35000, Rennes, France
natalia.grabar@univ-lille.fr, cyril.grouin@limsi.fr,
thierry.hamon@limsi.fr, vincent.claveau@irisa.fr

RÉSUMÉ

Les corpus textuels sont utiles pour diverses applications de traitement automatique des langues (TAL) en fournissant les données nécessaires pour leur création, adaptation ou évaluation. Cependant, dans certains domaines comme le domaine médical, l'accès aux données est rendu compliqué, voire impossible, pour des raisons de confidentialité et d'éthique. Il existe néanmoins de réels besoins en corpus cliniques pour l'enseignement et la recherche. Pour répondre à ce défi, nous présentons dans cet article le corpus *CAS* contenant des cas cliniques de patients, réels ou fictifs, que nous avons compilés. Ces cas cliniques en français couvrent plusieurs spécialités médicales et focalisent donc sur différentes situations cliniques. Actuellement, le corpus contient 4 300 cas (environ 1,5M d'occurrences de mots). Il est accompagné d'informations (discussions des cas cliniques, mots-clés, etc.) et d'annotations que nous avons effectuées au regard des besoins de la recherche en TAL dans ce domaine. Nous présentons également les résultats de premières expériences de recherche et d'extraction d'information qui ont été effectuées avec ce corpus annoté. Ces expériences peuvent fournir une *baseline* à d'autres chercheurs souhaitant travailler avec les données.

ABSTRACT

Annotated corpus with clinical cases in French.

Textual corpora are important for several NLP tasks because they provide suitable information for designing, adapting and evaluating these NLP applications. Yet, in some domains, such as the medical one, for confidentiality and ethical reasons, access to representative data is complicated or even impossible. Still, real need exists for this kind of corpora, both for training and research. In this paper, we propose the *CAS* corpus in French containing clinical cases of patients, real or fake. They cover various medical specialities and focuse on different clinical situations. Currently, the corpus contains 3,600 cases (almost 1.3M word occurrences). This corpus is associated with additional information (discussions of clinical cases, key-words...) and annotations that we produced to answer common research issues in this domain. We also present results from preliminary experiments of information retrieval and extraction performed on this corpus. These experiments can provide a *baseline* for the researchers interested in working with these data.

MOTS-CLÉS : Corpus clinique, cas clinique, annotations, catégorisation, extraction d'information.

KEYWORDS: Clinical corpus, clinical case, annotations, categorization, information extraction.

1 Introduction

Les corpus textuels sont utiles pour diverses tâches et applications du traitement automatique des langues (TAL) car ils fournissent les informations nécessaires pour la création, l'adaptation et l'évaluation des applications et d'outils. Cependant, dans certains domaines, pour des raisons de confidentialité et d'éthique, l'accès aux données représentatives devient très compliqué voire impossible. Les domaines du médical et du juridique relèvent de cette situation : dans le domaine juridique, l'information sur les procès et les délibérations reste confidentielle, tandis que dans le domaine médical les dossiers cliniques de patients sont aussi confidentielles car le secret médical doit être respecté. Dans les deux cas, les données ne peuvent pas être utilisées en dehors du cadre initial, en raison de la présence de données nominatives.

Notons que depuis plusieurs années déjà, les outils et méthodes d'anonymisation et de désidentification sont devenus disponibles et fournissent des résultats compétitifs (Ruch *et al.*, 2000; Sibanda & Uzuner, 2006; Uzuner *et al.*, 2007; Grouin & Zweigenbaum, 2013) en atteignant jusqu'à 90 % de précision et de rappel. Ces outils ont été développés pour traiter des textes en plusieurs langues et provenant de différents domaines. Leur exploitation pourrait donc aider les chercheurs à accéder aux données sensibles. Cependant, les données désidentifiées peuvent aussi être difficiles à obtenir et à utiliser pour la recherche car il a été noté que le risque de ré-identification des personnes persiste. Cela concerne par exemple les patients (Meystre *et al.*, 2014; Grouin *et al.*, 2015) dont les histoires médicales peuvent être uniques. D'autres difficultés d'ordre institutionnel ou juridique peuvent également complexifier la situation et l'accès aux données. Pour ces diverses raisons, la désidentification des données personnelles n'est souvent pas suffisante pour pouvoir les exploiter dans les contextes de recherche et d'enseignement en dehors des structures hospitalo-universitaires.

Néanmoins, il existe de réels besoins en développement de méthodes et outils visant les applications orientées sur des domaines spécialisés. Il en est ainsi dans le domaine médical, où des outils de recherche et d'extraction d'information sont nécessaires, par exemple pour le recrutement et l'inclusion de patients dans des essais cliniques, la recherche de patients similaires, le codage PMSI, etc. De manière plus fondamentale, il s'agit de tâches comme par exemple l'indexation des dossiers cliniques, l'étude de la temporalité, de l'incertitude ou de la négation, l'extraction des traitements prescrits ou des effets indésirables, etc. (Embi *et al.*, 2005; Hamon & Grabar, 2010; Uzuner *et al.*, 2011; Fletcher *et al.*, 2012; Sun *et al.*, 2013; Campillo-Gimenez *et al.*, 2015; Kang *et al.*, 2017). Ces questions de recherche sont communément abordées en langue anglaise, qui dispose de corpus dédiés, mais restent fragiles dans d'autres langues, comme le français, faute de corpus disponibles et accessibles pour la recherche.

Un autre point crucial, qui motive grandement notre travail, concerne la fiabilité des outils et la reproductibilité des résultats avec des données similaires provenant de sources différentes ou même avec des données provenant du même type de sources. Les travaux de recherche du domaine biomédical souffrent ainsi d'une vive critique en raison du manque de reproductibilité des résultats obtenus (Chapman *et al.*, 2011; Collins & Tabak, 2014; Cohen *et al.*, 2016). Une première étape vers la reproductibilité passe par la disponibilité d'outils, de corpus et de données de référence.

Dans ce travail, nous nous focalisons sur la création d'un corpus disponible contenant des données issues ou proches des données cliniques. L'objectif de cet article consiste à présenter le corpus de cas cliniques en français, les annotations actuellement disponibles et quelques premières expériences et leurs résultats. Dans ce qui suit, nous présentons d'abord quelques travaux sur la création de corpus médicaux en mettant l'accent sur les corpus disponibles pour la recherche (section 2). Nous décrivons

ensuite le corpus de cas cliniques en français (section 3) que nous proposons, les annotations actuelles (section 4) et les expériences qu'il a permis d'effectuer jusqu'ici (sections 5 à 7). Nous concluons en indiquant quelques directions de travaux futurs (section 8).

2 Corpus cliniques disponibles librement

Dans le domaine médical, nous pouvons distinguer deux principaux types de corpus : scientifiques et cliniques. Les *corpus scientifiques* proviennent de la littérature scientifique. Ils deviennent de plus en plus disponibles pour la recherche grâce aux initiatives de publications ouvertes, comme celles soutenues par la NLM (National Library of Medicine) dans le portail PUBMED [1] spécifiquement dédié au domaine biomédical, ou des portails généralistes comme HAL [2] et ISTEX [3]. Certains corpus scientifiques fournissent des annotations et catégorisations précises. Ils sont souvent créés pour des compétitions TAL (Kelly *et al.*, 2013; Goeuriot *et al.*, 2014) ou proviennent des travaux de chercheurs (Tsuruoka *et al.*, 2005; Szarvas *et al.*, 2008).

En ce qui concerne les *corpus cliniques*, ils sont liés aux événements cliniques des patients (histoire médicale, soins médicaux, prescriptions, analyses de laboratoires, procédures chirurgicales, etc.). Il est compliqué d'avoir un accès libre à ce type de données pour des raisons évoquées plus haut (données nominales et sensibles, risque de ré-identification, contexte institutionnel...).

Le présent article s'intéresse à ce dernier type de corpus et notre revue de la littérature porte sur les corpus cliniques librement disponibles pour la recherche :

— Le corpus *MIMIC* (Medical Information Mart for Intensive Care), actuellement dans sa troisième version, fournit le plus grand ensemble de données cliniques, structurées et non structurées, en anglais. MIMIC III provient d'une seule institution et contient les informations relatives aux patients qui y sont admis. Ces données concernent les examens médicaux, médicaments prescrits, résultats de laboratoire, encodage des actes et des diagnostics, rapports d'imagerie, durée de séjour à l'hôpital, etc. Ce corpus est exploité dans de nombreuses applications académiques et industrielles, pour la recherche, pour l'amélioration des soins et pour l'enseignement (Johnson *et al.*, 2016), satisfaisant ainsi toute la palette des contextes propres au domaine biomédical. Plusieurs travaux de recherche utilisent ces données pour la prédiction de la mortalité (Anand *et al.*, 2018; Feng *et al.*, 2018), l'identification du diagnostic et le codage (Perotte *et al.*, 2014; Li *et al.*, 2018), l'étude de la temporalité (Che *et al.*, 2018) ou encore la recherche de cas similaires (Gabriel *et al.*, 2018). Les données de ce corpus ont notamment été utilisées dans plusieurs compétitions de TAL, dont nous décrivons plusieurs ici : I2B2, N2C2, CLEF-eHEALTH.

— *I2B2* (Informatics for Integrating Biology and the Bedside) [4] est une compétition dont l'objectif est de motiver le développement et l'évaluation d'outils du TAL sur les données cliniques. Les données exploitées sont en anglais et désidentifiées. Les différentes éditions ont proposé des annotations spécifiques sur la désidentification, l'identification de fumeurs, les informations liées aux médicaments, les relations sémantiques entre entités, ou la temporalité (Uzuner, 2008; Uzuner *et al.*, 2011; Sun *et al.*, 2013).

1. https://www.ncbi.nlm.nih.gov/pubmed
2. https://hal.archives-ouvertes.fr/
3. https://www.istex.fr/
4. https://www.i2b2.org/NLP/DataSets/Main.php

— *N2C2* (National NLP Clinical Challenges) [5] a lieu depuis 2018. La compétition porte par exemple sur l'inclusion de patients dans les essais cliniques, la détection des effets indésirables provoqués par le prise de médicaments, la détection de similarités textuelles, l'extraction de l'histoire familiale de maladies ou la normalisation de concepts . Cette compétition a pris le relais de I2B2, tout en proposant des tâches plus complexes et plus ancrées dans la réalité et l'activité clinique.
— *CLEF-eHEALTH* [6] a connu plusieurs éditions : la détection de maladies et la normalisation des abréviations en 2013 et 2014, le traitement des notes d'infirmiers australiens en 2016, l'extraction des causes de décès dans les certificats de décès en français issus du CépiDc [7] en 2016 et 2017.
— Le défi *eHealth-KD* 2019 [8] vise à modéliser la langue utilisée dans les documents cliniques en espagnol et à traiter automatiquement ces documents. Deux tâches sont proposées : identifier et classer des séquences clés, puis détecter les relations sémantiques entre séquences.

Finalement, les données médicales proches des données cliniques peuvent aussi être trouvées dans les protocoles d'essais cliniques. Des exemples de ce type de corpus comportent les annotations d'informations sur les valeurs numériques en anglais (Claveau *et al.*, 2017), et de négation en français et portugais brésilien (Dalloux *et al.*, 2018).

3 Corpus de cas cliniques en français

Nous proposons le corpus nommé *CAS* qui contient des cas cliniques rédigés en français. Les cas cliniques décrivent les situations cliniques de patients, réels désidentifiés ou fictifs. Ils sont publiés dans différentes sources de données (scientifique, didactique, associatif, juridique...). Ils sont anonymisés au moment de la publication. Les cas cliniques ont pour objectif de présenter des situations cliniques typiques, dans un objectif didactique, ou des situations rares et complexes (propriété rencontrée dans des cadres scientifique et juridique). La figure 1 présente un exemple de cas clinique. Nous observons que les informations fournies sont de nature diverse : genre et âge du patient, motif de la consultation ou de l'hospitalisation, observation et résultats d'examens cliniques, résultats d'examens biologiques, traitements effectués (traitements chirurgicaux dans l'exemple de la figure), évolution de la maladie. En ceci, le contenu des cas cliniques est vraiment très proche du contenu des dossiers cliniques et en offre donc un bon exemple.

Une première version du corpus a été présentée dans une publication antérieure (Grabar *et al.*, 2018). Depuis, le corpus a été fondamentalement enrichi. Actuellement, le corpus global contient pas loin de 4 300 cas, soit presque 1 500 000 occurrences de mots. Le contenu provient de différentes sources (littérature scientifique, matériel didactique, support des associations, affaires juridiques) et représente différentes spécialités médicales (cardiologie, urologie, oncologie, obstétrique, pneumologie, gastro-entérologie, gériatrie, pharmacologie, etc.). En fonction de la spécialité, l'accent est mis sur des aspects différents (diagnostic d'une maladie, prise en charge, intervention chirurgicale, interactions médicamenteuses, etc.) et les cas peuvent aussi bien relater toute l'histoire de la maladie des patients que de se focaliser sur un épisode donné. Les cas recensés ont été publiés dans différents pays francophones (France, Belgique, Suisse, Canada, pays africains, pays tropicaux, etc.). Il s'agit donc

5. https://n2c2.dbmi.hms.harvard.edu/
6. https://sites.google.com/site/shareclefehealth/
7. http://www.cepidc.inserm.fr/
8. https://knowledge-learning.github.io/ehealthkd-2019

B.A., âgé de 36 ans, sans antécédents notables, a été admis en février 1994 pour des douleurs lombaires droites évoluant dans un contexte d'altération de l'état général. L'examen clinique avait montré une tension artérielle à 10/06 mm Hg chez ce patient apyrétique, avec un examen abdominal et neurologique normal par ailleurs. Les examens biologiques montraient un taux de globules blancs à 7000/mm3 , une créatinine à 8 mg/l et une glycémie à 0,90 g/l. L'abdomen sans préparation ne montrait pas de calcifications et l'échographie abdominale avait montré une masse latéro-vertébrale droite refoulant le rein droit vers l'extérieur (Figure 1). La tomodensitométrie abdominale (Figures 2 et 3) avait objectivé une formation tissulaire isodense arrondie de 5 cm de diamètre située en plein parenchyme du muscle psoas droit. Une biopsie échoguidée de la tumeur n'avait pas ramené de tissu tumoral. L'intervention menée par une lombotomie avait découvert une tumeur encapsulée, bien limitée de 5 cm de grand diamètre incluse dans le muscle psoas. Une tumorectomie complète était réalisée. A la coupe, la tumeur présentait un aspect blanchâtre fasciculé, de consistance ferme. A l'examen microscopique, on avait trouvé une prolifération de cellules fibroblastiques fusiformes sans anomalies cytologiques agencées en faisceaux dissociés par l'oedème et du tissu conjonctif comportant des petits capillaires, concluant à un fibrome. L'évolution a été bonne avec un recul de 4 ans.

FIGURE 1 – Un exemple de cas clinique (les références à des figures font partie du document)

de productions effectuées en français et décrivant des situations cliniques assez typiques et réelles de patients susceptibles de venir en consultation ou en hospitalisation dans un hôpital francophone. Les cas cliniques sont écrits par les médecins : les mêmes personnes qui écrivent les dossiers hospitaliers des patients.

Par ailleurs, les cas peuvent bénéficier de différents types d'annotations, comme présenté dans la section 4, et être associés avec d'autres types d'information. Les informations associées dépendent des sources d'où proviennent les cas. Par exemple, les cas cliniques publiés dans la littérature scientifique sont souvent accompagnés d'une discussion et des mots-clés, les cas cliniques provenant du matériel didactique peuvent être accompagnés de questions de contrôle des connaissances, alors que les cas provenant des affaires juridiques sont typiquement associés avec les jugements et pénalités.

4 Annotations du corpus

Les annotations que nous présentons concernent un sous-ensemble du corpus composé de 717 cas cliniques (soit 232 000 occurrences de mots). Comme l'ensemble du corpus, ces cas couvrent différentes spécialités médicales et proviennent de plusieurs sources. Ce corpus de 717 cas cliniques a été mis à disposition de la compétition DEFT 2019 [9]. En dehors des annotations présentées plus bas (les informations démographiques (section 4.1) et les informations cliniques générales (section 4.2)), les cas sont annotés manuellement avec des informations sémantiques plus fines (maladies, signes et symptômes, médicaments, procédures, dates, examens cliniques et biologiques, etc.) et annotés automatiquement avec des informations linguistiques (étiquetage morpho-syntaxique). Nous faisons également le bilan quantitatif des annotations démographiques et cliniques générales (section 4.3).

9. https://deft.limsi.fr/2019/

4.1 Informations démographiques

Les informations démographiques couvrent l'*âge* et le *genre* des patients. Les portions textuelles permettant d'en déterminer les valeurs sont annotées : la valeur numérique et l'unité pour les âges *(2 mois et demi, 36 ans, la quarantaine)*, les valeurs réelles *(sexe féminin, garçon)* ou les indices linguistiques permettant de les inférer : participes passés *(hospitalisé, intubée)*, pronoms personnels ou démonstratifs, déclencheurs *(M., Mme)*, expressions *(le patient, cette patiente)*. Les valeurs obtenues sont normalisées sous la forme d'un entier pour l'âge et des valeurs "féminin" ou "masculin" pour le genre (il n'existe aucun cas d'hermaphrodisme ou de dysgénésie).

4.2 Informations cliniques générales

Les informations cliniques générales concernent l'*origine* de la consultation (pathologie, signe ou symptôme qui se trouvent à l'origine de la consultation ou de l'hospitalisation décrites dans le cas) et l'état du patient à l'*issue* de l'hospitalisation (guérison, amélioration, stabilité, détérioration, décès). Lorsque le cas clinique intègre l'histoire de la maladie avec plusieurs épisodes d'hospitalisations ou de consultations, c'est le dernier épisode qui est retenu comme origine de la consultation ou de l'hospitalisation décrite et par rapport à laquelle une issue peut être définie.

4.3 Statistiques

Classe	*Annotateur 1/Annotateur 2*	*Annotateur 1/consensus*	*Annotateur 2/consensus*
âge	0,9844	0,9887	0,9944
genre	0,8044	0,9903	0,8143
issue	0,4654	0,6204	0,8152
origine	0,8734	0,8886	0,9755

TABLE 1 – Accords inter-annotateurs (F-mesure) calculés avec BRATeval : comparaison des portions pour *âge* et *origine*, et des valeurs normalisées pour *genre* et *issue*.

Le corpus a été annoté par deux annotateurs de manière indépendante. Le tableau 1 fournit les accords inter-annotateurs (F-mesure) calculés avec l'outil BRATeval. L'évaluation porte sur la portion annotée pour les classes *âge* et *origine*, et sur les valeurs normalisées pour les classes *genre* (valeurs possibles : masculin, féminin) et *issue* (valeurs possibles : guérison, amélioration, stable, détérioration, décès). Un consensus a permis de corriger les erreurs et oublis d'annotations. En cas de désaccords sur les frontières, qui concernaient la classe *origine*, la portion la plus englobante est conservée. Les désaccords sur l'issue concernent des valeurs proches (guérison/amélioration, stable/amélioration), des oublis d'annotation, ou des absences volontaires d'annotation dues à la difficulté de choisir la bonne valeur. Nous observons que la classe *issue* est moins simple qu'il n'y paraît, suscitant de nombreuses discussions lors du consensus. Globalement, nous pouvons voir que : (1) l'accord entre les deux annotateurs est proche de la perfection pour l'âge (0,9844), (2) l'accord est très bon pour le genre et l'origine (0,8044 et 0,8734), et (3) l'accord est faible pour l'issue (0,4654). Pour cette dernière catégorie, les valeurs de l'accord des deux annotateurs par rapport au consensus indiquent que chacun des annotateurs a fait des erreurs ou omissions d'annotation, de même que des annotations correctes qui ont été retenues dans la version consensuelle de l'annotation.

Classe	*Valeurs normalisées et nombre d'occurrences*
âge	0-9 ans (56), 10-19 ans (63), 20-29 ans (100), 30-39 ans (109), 40-49 ans (99), 50-59 ans (132), 60-69 ans (75), 70-79 ans (54), 80-89 ans (14), 90-99 ans (4), âge inconnu (21)
genre	féminin (321), masculin (418)
issue	guérison (227), amélioration (256), stabilité (55), détérioration (23), décès (117)

TABLE 2 – Statistiques d'annotations du corpus de cas cliniques

Le tableau 2 donne la répartition des valeurs normalisées des classes *âge*, *genre* et *issue* dans le corpus. Certaines catégories (80-89 ans et 90-99 ans pour l'âge, et détérioration pour l'issue) sont sous-représentées par rapport à d'autres.

Grâce à ses annotations et ses informations, le corpus CAS permet de tester des applications utiles dans le domaine clinique, comme la catégorisation et l'extraction d'information. Nous avons défini trois tâches sur la base des annotations produites et des informations disponibles dans le corpus :

1. association des mots-clés avec les cas cliniques (section 5),
2. association des cas cliniques et des discussions (section 6),
3. extraction d'information clinique (section 7).

Pour chacune de ces tâches, nous proposons des techniques simples, se voulant des systèmes permettant de fournir des résultats initiaux (*baseline*). Ces systèmes se veulent simples dans leur conception (techniques bien connues) et dans la mesure où ils n'utilisent pas de connaissances externes. Elles sont présentées dans les sections suivantes.

5 Association des mots-clés avec les cas cliniques

La première tâche consiste à associer des mots-clés à chacun des cas cliniques. Avec le développement des systèmes d'information hospitalière, la recherche de dossiers cliniques devient un réel défi pour les praticiens qui désirent trouver un dossier particulier ou un patient donné dans la masse des informations existantes dans un hôpital. L'indexation des dossiers cliniques s'impose alors comme une étape préalable à la recherche d'information.

L'entrée de cette tâche est un ensemble de cas cliniques avec leurs discussions, un ensemble des mots-clés possibles et le nombre de mots-clés attendus. En effet, dans le contexte clinique, l'indexation ou le codage de dossiers médicaux sont souvent effectués de manière contrôlée en exploitant des terminologies médicales existantes. Dans notre tâche, la vérité-terrain, ou les données de référence, est constituée avec les mots-clés assignés par les auteurs eux-mêmes aux publications scientifiques qu'ils ont écrites et d'où proviennent les cas cliniques. Un mot-clé peut être associé à plusieurs cas cliniques, un cas cliniques peut recevoir un à plusieurs mots-clés, et certains mots-clés de l'ensemble ne doivent pas être associés aux cas cliniques ou leurs discussions.

Nous avons 290 cas cliniques/discussions dans le jeu d'entraînement et 213 cas cliniques/discussions dans le jeu de test. Dans les deux cas, l'ensemble de mots-clés regroupe les mots-clés de l'entraînement et du test et contient 1 311 mots-clés.

5.1 Évaluation

Nous avons constitué des données d'entraînement (290 cas avec leurs mots-clés), pour permettre l'emploi de méthodes d'apprentissage, et de test (213 cas). L'évaluation de cette tâche prend en compte la possibilité de produire une liste ordonnée de mots-clés candidats, du plus pertinent au moins pertinent. Pour confronter cette liste ordonnée à la liste de référence, nous utilisons deux mesures classiquement utilisées en Recherche d'Information : la moyenne des R-Précisions (précision mesurée au rang N, $Pr(N)$, où N est le nombre de mots-clés attendus pour ce cas clinique) et la MAP (*Mean Average Precision*, moyenne de l'*Average Precision* ; voir formule 1).

$$MAP = \frac{1}{|C|} \sum_{c \in \mathcal{C}} \frac{\sum_{k=1}^{m} Pr(k) * \mathbb{1}_{t(k) \in Rel(c)}}{N} \quad (1)$$

5.2 Systèmes de référence

A titre de comparaison, nous avons produit deux systèmes de référence, ou des *baselines*. Le tableau 3 indique les performances de ces systèmes de référence.

Système	*MAP*	*R-précision*
Baseline 1	0,177	0,236
Baseline 2	0,434	0,428

TABLE 3 – Résultats des deux *baselines* pour la tâche 1 d'association de mots-clés aux cas cliniques et leurs discussions.

Un premier système de référence consiste à rechercher les mots-clés de la liste à l'identique dans chaque couple cas clinique/discussion, puis à sélectionner les mots-clés dont la fréquence d'utilisation dans le couple cas clinique/discussion est la plus élevée. En cas de fréquences identiques entre plusieurs mots-clés (par exemple, une fréquence de 1), nous conservons les mots-clés les plus longs, en émettant l'hypothèse qu'un mot-clé long est plus significatif qu'un mot-clé court. Enfin, nous limitons le nombre de mots-clés retournés au nombre de mots-clés attendu. Notons que cette approche donne la possibilité d'exploiter les cas cliniques et/ou la discussion. Les résultats indiquent que le traitement séparé du cas et de la discussion, avec une sélection des mots-clés a posteriori, est plus efficace que la fusion des deux. Cette *baseline* obtient une MAP de 0,177 et une R-précision de 0,236 sur les données de test.

Le deuxième système de référence exploite la pondération Okapi-BM25 (Robertson *et al.*, 1998) pour ordonner les candidats mots-clés. Cette pondération permet ainsi de tenir compte de la fréquence des mots-clés dans le cas clinique traité, mais aussi dans l'ensemble des cas cliniques. Tous les mots-clés fournis sont cherchés dans les documents et ceux identifiés sont pondérés par BM-25. La liste retournée correspond ainsi aux mots-clés trouvés, ordonnée par le score BM-25 décroissant. Cette approche obtient une MAP de 0,434 et une R-précision de 0,428 sur les données de test.

6 Association des cas cliniques et des discussions

La deuxième tâche consiste à associer la discussion au cas clinique correspondant, ce qui peut se révéler utile pour les médecins qui veulent identifier dans la littérature scientifique des observations cliniques similaires à celles de leurs patients. Une telle recherche bibliographique vise à trouver les méthodes de diagnostic ou de traitement les plus appropriées. L'entrée de cette tâche est un ensemble de cas cliniques et un ensemble de discussions. Chaque cas doit être associé à une discussion. Une discussion donnée peut être associée à plus d'un cas clinique.

Nous avons 290 cas (et leurs discussions) dans le jeu d'entraînement et 213 cas (et leurs discussions) dans le jeu de test. Il existe donc des doublons au sein des discussions.

6.1 Évaluation

Pour cette tâche, une seule réponse est attendue : une seule discussion à associer à un cas clinique. En revanche, comme indiqué plus haut, une même discussion peut concerner plusieurs cas. L'évaluation se fait classiquement par les mesures du rappel et de la précision, calculées globalement (c'est-à-dire, formellement, macro-précision et macro-rappel) : si le système renvoie une réponse pour tous les cas, ces deux mesures sont égales. Le script d'évaluation gère les doublons qui se trouvent au sein des discussions : il suffit qu'une discussion, parmi les discussions doublons correctes, soit associée à un cas clinique.

6.2 Systèmes de référence

Système	*Précision*	*Rappel*
Baseline	0,9500	0,9500

TABLE 4 – Résultats de la *baseline* pour la tâche 2 d'association des cas cliniques et des discussions.

L'approche *baseline* que nous proposons consiste à calculer les similarités entre toutes les discussions et tous les cas cliniques. Ces derniers sont simplement représentés comme des sacs-de-mots. La similarité utilisée est de nouveau Okapi-BM25. On obtient ainsi une matrice de similarité entre cas et discussions. A ce point, une discussion peut être plus proche du cas c_1 que que du cas c_2, en terme du score BM-25, et être la première classée pour c_2 et la cinquième pour c_1. L'attribution optimale se fait alors en utilisant l'algorithme hongrois (Kuhn & Yaw, 1955) sur cette matrice de similarités. La précision (équivalente au rappel) obtenue par cette approche, sur le jeu de test, est de 0,95 (tableau 4).

7 Extraction d'information clinique

Les annotations manuelles disponibles sur ce corpus fournissent la possibilité d'effectuer d'autres expériences qui sont également proches des besoins cliniques en traitement d'informations. Il s'agit typiquement de l'extraction d'information pour la recherche de patients avec un profil donné ou par rapport aux critères d'inclusion dans les protocoles d'essais cliniques. Pour cette expérience, quatre informations sont annotées et recherchées : l'âge, le genre, l'issue et la portion de texte expliquant

la raison d'admission du patient. L'entrée de cette tâche est un ensemble de cas cliniques. Un cas clinique concerne en général un patient mais certains cas peuvent être concernés par plus d'un patient. Il n'est pas nécessaire d'associer les informations extraites (par exemple, l'âge et le genre) entre elles.

Le jeu d'entraînement contient 290 cas cliniques et le jeu de test 427 cas cliniques. Chaque cas est en général annoté avec les quatre types d'information.

7.1 Évaluation

Pour cette tâche, les quatre types d'informations à extraire sont évalués selon deux protocoles, en fonction de la nature de l'information :

— L'âge, le genre, et l'issue sont classiquement évalués par la précision et le rappel.

— L'admission est représentée par une portion de texte ou, dans quelques cas, par plusieurs portions de texte. Pour comparer la portion attendue à celle(s) prédite(s), nous utilisons plusieurs mesures. Nous calculons les valeurs de rappel et précision sur les mots de ces portions de textes. Ces mesures peuvent être faites au niveau de chaque cas et moyennées, ou calculées globalement sur l'ensemble des cas, résultats en micro-précision et micro-rappel, ou macro-précision et macro-rappel. Nous proposons également de mesurer l'intersection en nombre de mots entre la portion attendue et la portion prédite, normalisée par l'ensemble des mots de la référence et de la prédiction. Cette mesure effectuée pour chaque cas est ensuite moyennée, résultant ainsi dans la mesure que nous appelons *micro-overlap*, définie formellement dans la formule 2.

$$micro - overlap = \frac{1}{|C|} \sum_{c \in \mathcal{C}} \frac{\sum_{k=1}^{m} \frac{TP}{TP + FP + FN}}{N} \tag{2}$$

7.2 Systèmes de référence

Nous avons produit deux systèmes de référence. Les résultats obtenus par ces deux systèmes se trouvent dans le tableau 5.

	Système à base de règles		Apprentissage supervisé	
Classe	*Précision*	*Rappel*	*Précision*	*Rappel*
âge	0,7897	0,7685	0,9608	0,9116
genre	0,9138	0,9014	0,9602	0,9535
issue	0,4444	0,4247	0,5321	0,5246
origine (micro)	0,4182	0.0061	0,7707	0,5559
origine (macro)	0,0321	0.0163	0,5141	0,5647

TABLE 5 – Résultats des méthodes de baseline pour la tâche 3 : extraction de l'âge, du genre, de l'issue, et de l'origine d'admission

Un premier système de référence repose sur un ensemble limité de règles propres à chaque catégorie : 5 règles pour le genre, 9 pour l'âge, et 7 pour l'issue. Il exploite pour ceci une liste de termes

(*femme, homme, madame, monsieur...*), les parties anatomiques genrées, les pronoms personnels pour compléter l'identification du genre. Sur la catégorie *Origine*, ce système est limité à seulement six règles pour traiter les portions commençant par la préposition "pour" suivi de termes ressemblant à des signes ou symptômes (*pour des épisodes fugaces de palpitations*, *pour une gêne respiratoire*, etc.). Ce travail limité et rapide ne peut donner cependant lieu à des résultats viables sur cette dernière catégorie. Comme indiqué dans le tableau 5, ce système a des performances élevées pour le genre et l'âge. En revanche les performances des deux autres catégories restent basses, surtout en ce qui concerne le rappel.

Un deuxième système de référence proposé exploite des approches par apprentissage artificiel. Deux approches sont exploitées :

— Le genre et l'issue sont considérés comme des problèmes de catégorisation de texte. Nous utilisons un algorithme de Régression Logistique dans lequel nous représentons le texte sous la forme de sac-de-mots, avec une simple pondération TF *(term frequency)*. Le modèle est appris sur le jeu d'entraînement et utilisé ensuite pour prédire le genre et l'issue pour les cas du jeu de test.

— Pour l'âge et l'admission, il s'agit de repérer dans les documents les portions faisant mention de l'âge et de la raison d'admission. Ils ont donc été considérés comme des problèmes d'étiquetage. Les textes sont étiquetés en parties-du-discours et lemmatisés avec TagEx [10]. Pour l'entraînement, les informations d'âge et d'admission sont projetés sur le document sous la forme d'étiquette IOB. Nous entraînons ensuite un modèle CRF (implémenté par Wapiti (Lavergne *et al.*, 2010)) sur ces données qui est ensuite appliqué au données du jeu de test.

Les résultats obtenus sont indiqués dans le tableau 5. Nous voyons que deux catégories (âge et genre) montrent des performances très élevées, étant supérieures à 0,90 en termes de précision et de rappel. Les deux autres catégories ont des performances un peu plus modestes mais qui restent élevées : entre 0,50 et 0,77. Pour ce système aussi, le rappel est plus difficile à gérer que la précision.

8 Conclusion

Nous avons décrit un corpus de cas cliniques en français, qui correspondent à des données proches de celles créées et utilisées dans le contexte hospitalier. Actuellement, le corpus contient 4 300 cas cliniques (environ 1,5M d'occurrences de mots). Une partie du corpus (717 cas cliniques) a été annotée avec quatre types d'informations sémantiques (âge et genre du patient, origine de consultation et issue de consultation). L'accord inter-annotateurs, calculé avec la F-mesure, est supérieur à 0,80 pour trois catégories (âge, genre et origine) et est de 0,4654 pour la catégorie issue. Cette dernière a en effet présenté des difficultés d'annotation. De plus, ces 717 cas cliniques sont également associés avec les mots-clés et une discussion, tous les deux étant fournis par les publications d'origine.

Le corpus de cas cliniques de patients décrit dans cet article peut être utilisé pour l'enseignement et la recherche. Ainsi, plusieurs cadres d'évaluation de tâches de catégorisation et d'extraction d'information sont en cours de développement, montrant ainsi le potentiel que CAS représente pour la recherche. La mise à disposition de ce corpus pour la compétition DEFT 2019 [11] en fait partie,

10. TagEx est un outil d'étiquetage morpho-syntaxique et de lemmatisation développé à l'IRISA et disponible en web-service : `https://allgo.inria.fr`. Il est adapté au traitement de documents du domaine biomédical.

11. (`https://deft.limsi.fr/2019/`

alors que les résultats des systèmes de référence ont vocation de fournir des données de comparaison par rapport auxquelles d'autres systèmes pourront se positionner.

De manière plus générale, nous pensons que la disponibilité de ce corpus et des annotations vont stimuler la recherche sur les données de type clinique en langue française. Ceci va contribuer à garantir la reproductibilité des résultats et la robustesse des méthodes et outils.

Nous prévoyons d'enrichir le corpus avec d'autres cas cliniques et de fournir d'autres annotations consensuelles. Ce corpus et ses annotations pourront donc faire objet d'autres compétitions TAL.

Remerciements

Ce travail a bénéficié d'une aide de l'État attribuée au labex COMIN LABS et gérée par l'Agence Nationale de la Recherche au titre du programme « Investissements d'avenir » portant la référence ANR-10-LABX-07-01. Ce travail s'inscrit également 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

ANAND R., STEY P., JAIN S., BIRON D., BHATT H., MONTEIRO K., FELLER E., ML R., IN S. & ES C. (2018). Predicting mortality in diabetic icu patients using machine learning and severity indices. In *AMIA Jt Summits Transl Sci Proc*, p. 310–319.

CAMPILLO-GIMENEZ B., BUSCAIL C., ZEKRI O., LAGUERRE B., LE PRISÉ E., DE CREVOISIER R. & CUGGIA M. (2015). Improving the pre-screening of eligible patients in order to increase enrollment in cancer clinical trials. *Trials*, **16**(1), 1–15.

CHAPMAN W. W., NADKARNI P. M., HIRSCHMAN L., D'AVOLIO L. W., SAVOVA G. K. & UZUNER O. (2011). Overcoming barriers to nlp for clinical text : the role of shared tasks and the need for additional creative solutions. *J Am Med Inform Assoc*, **18**(5), 540–543.

CHE Z., PURUSHOTHAM S., CHO K., SONTAG D. & LIU Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Sci Rep*, **8**(1), 6085.

CLAVEAU V., SILVA OLIVEIRA L. E., BOUZILLÉ G., CUGGIA M., CABRAL MORO C. M. & GRABAR N. (2017). Numerical eligibility criteria in clinical protocols : annotation, automatic detection and interpretation. In *AIME (Artifical Intelligence in Medicine in Europe)*.

COHEN K. B., XIA J., ROEDER C. & HUNTER L. E. (2016). Reproducibility in natural language processing : A case study of two r libraries for mining pubmed/medline. In *LREC Int Conf Lang Resour Eval*, p. 6–12.

COLLINS F. & TABAK L. (2014). Nih plans to enhance reproducibility. *Nature*, **505**, 612–613.

DALLOUX C., CLAVEAU V., GRABAR N. & MORO C. (2018). Portée de la négation : détection par apprentissage supervisé en français et portugais brésilien. In *TALN 2018*, p. 1–6.

EMBI P., JAIN A., CLARK J. & HARRIS C. (2005). Development of an electronic health record-based clinical trial alert system to enhance recruitment at the point of care. In *Ann Symp Am Med Inform Assoc (AMIA)*, p. 231–35.

FENG M., MCSPARRON J., KIEN D., STONE D., ROBERTS D., SCHWARTZSTEIN R., VIEILLARD-BARON A. & CELI L. (2018). Transthoracic echocardiography and mortality in sepsis : analysis of the mimic-iii database. *Intensive Care Med*, **44**(6), 884–892.

FLETCHER B., GHEORGHE A., MOORE D., WILSON S. & DAMERY S. (2012). Improving the recruitment activity of clinicians in randomised controlled trials : A systematic review. *BMJ Open*, **2**(1), 1–14.

GABRIEL R., KUO T., MCAULEY J. & HSU C. (2018). Identifying and characterizing highly similar notes in big clinical note datasets. *J Biomed Inform*, **82**, 63–69.

GOEURIOT L., KELLY L., LI W., PALOTTI J., PECINA P., ZUCCON G., HANBURY A., JONES G. & MÜLLER H. (2014). Share/clef ehealth evaluation lab 2014, task 3 : User-centred health information retrieval. In *CLEF*, Lecture Notes in Computer Science (LNCS), p. 43–61 : Springer.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). Cas : French corpus with clinical cases. In *LOUHI 2018*, p. 1–12, Bruxelles, Belgique.

GROUIN C., GRIFFON N. & NÉVÉOL A. (2015). Is it possible to recover personal health information from an automatically de-identified corpus of French EHRs ? In *Proc of LOUHI*, Lisbon, Portugal.

GROUIN C. & ZWEIGENBAUM P. (2013). Automatic de-identification of french clinical records : Comparison of rule-based and machine-learning approches. In *Stud Health Technol Inform, Proc of MedInfo*, volume 192, p. 476–80, Copenhagen, Denmark.

HAMON T. & GRABAR N. (2010). Linguistic approach for identification of medication names and related information in clinical narratives. *J Am Med Inform Assoc*, **17**(5), 549–54.

JOHNSON A. E., POLLARD T. J., SHEN L., WEI H. LEHMAN 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), 1–9.

KANG T., ZHANG S., TANG Y., HRUBY G. W., RUSANOV A., ELHADAD N. & WENG C. (2017). EliIE : An open-source information extraction system for clinical trial eligibility criteria. *J Am Med Inform Assoc*, **24**(6), 1062–1071.

KELLY L., GOEURIOT L., SUOMINEN H., MOWERY D. L., VELUPILLAI S., CHAPMAN W. W., ZUCCON G. & PALOTTI J. (2013). Overview of the share/clef ehealth evaluation lab 2013. In *CLEF*, Lecture Notes in Computer Science (LNCS) : Springer.

KUHN H. W. & YAW B. (1955). The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, **2**, 83–97.

LAVERGNE T., CAPPÉ O. & YVON F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 504–513 : Association for Computational Linguistics.

LI M., FEI Z., ZENG M., WU F., LI Y., PAN Y. & WANG J. (2018). Automated ICD-9 coding via a deep learning approach. In *IEEE/ACM Trans Comput Biol Bioinform*.

MEYSTRE S., SHEN S., HOFMANN D. & GUNDLAPALLI A. (2014). Can physicians recognize their own patients in de-identified notes ? In *Stud Health Technol Inform 205*, p. 778–82.

PEROTTE A., PIVOVAROV R., NATARAJAN K., WEISKOPF N., WOOD F. & ELHADAD N. (2014). Diagnosis code assignment : models and evaluation metrics. *J Am Med Inform Assoc*, **21**, 231–237.

ROBERTSON S. E., WALKER S. & HANCOCK-BEAULIEU M. (1998). Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive. In *Proc. of the 7th Text Retrieval Conference, TREC-7*, p. 199–210.

RUCH P., BAUD R. H., RASSINOUX A.-M., BOUILLON P. & ROBERT G. (2000). Medical document anonymization with a semantic lexicon. In *Ann Symp Am Med Inform Assoc (AMIA)*, p. 729–733, Los Angeles, CA.

SIBANDA T. & UZUNER O. (2006). Role of local context in de-identification of ungrammatical, fragmented test. In *NAACL-HLT 2006*, New York, USA.

SUN W., RUMSHISKY A. & UZUNER Ö. (2013). Evaluating temporal relations in clinical text : 2012 i2b2 challenge. *JAMIA*, **20**(5), 806–813.

SZARVAS G., VINCZE V., FARKAS R. & CSIRIK J. (2008). The BioScope corpus : annotation for negation, uncertainty and their scope in biomedical texts. In *BIONLP*, p. 38–45.

TSURUOKA Y., TATEISHI Y., KIM J., OHTA T., MCNAUGHT J., ANANIADOU S. & TSUJII J. (2005). Developing a robust part-of-speech tagger for biomedical text. *LNCS*, **3746**, 382–392.

UZUNER O. (2008). Second i2b2 workshop on natural language processing challenges for clinical records. In *Ann Symp Am Med Inform Assoc (AMIA)*, p. 1252–3.

UZUNER O., LUO Y. & SZOLOVITS P. (2007). Evaluating the state-of-the-art in automatic de-identification. *J Am Med Inform Assoc*, **14**, 550–563.

UZUNER O., SOUTH B. R., SHEN S. & DUVALL S. L. (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*, **18**(5), 552–556.

Curriculum d'apprentissage : reconnaissance d'entités nommées pour l'extraction de concepts sémantiques

Antoine Caubrière[1] Natalia Tomashenko[2] Yannick Estève[2]
Antoine Laurent[1] Emmanuel Morin[3]
(1) LIUM, Avenue Olivier Messiaen, 72085 Le Mans, France
(2) LIA, 339 Chemin des Meinajaries, 84140 Avignon, France
(3) LS2N, 2 Chemin de la Houssinière, 44322 Nantes, France
prénom.nom@{univ-lemans,univ-avignon,univ-nantes}.fr

RÉSUMÉ

Dans cet article, nous présentons une approche de bout en bout d'extraction de concepts sémantiques de la parole. En particulier, nous mettons en avant l'apport d'une chaîne d'apprentissage successif pilotée par une stratégie de curriculum d'apprentissage. Dans la chaîne d'apprentissage mise en place, nous exploitons des données françaises annotées en entités nommées que nous supposons être des concepts plus génériques que les concepts sémantiques liés à une application informatique spécifique. Dans cette étude, il s'agit d'extraire des concepts sémantiques dans le cadre de la tâche MEDIA. Pour renforcer le système proposé, nous exploitons aussi des stratégies d'augmentation de données, un modèle de langage 5-gramme, ainsi qu'un mode étoile aidant le système à se concentrer sur les concepts et leurs valeurs lors de l'apprentissage. Les résultats montrent un intérêt à l'utilisation des données d'entités nommées, permettant un gain relatif allant jusqu'à 6,5 %.

ABSTRACT

Curriculum learning : named entity recognition for semantic concept extraction

In this paper, we present an end-to-end approach for semantic concept extraction from speech. In particular, we highlight the contribution of a successive learning chain driven by a curriculum learning strategy. In the learning chain, we use French data with named entity annotations that we assume are more generic concepts than semantic concept related to a specific computer application. In this study, the aim is to extract semantic concept as part of the MEDIA task. To improve the proposed system, we also use data augmentation, 5-gram langage model and a star mode to help the system focus on concepts and their values during the training. Results show an interest in using named entity data, allowing a relative gain up to 6.5%.

MOTS-CLÉS : Curriculum d'apprentissage, transfert d'apprentissage, bout en bout, extraction de concepts sémantiques, entités nommées.

KEYWORDS: Curriculum learning, transfer learning, end-to-end, semantic concept extraction, named entity.

1 Introduction

L'apprentissage humain est réalisé par étapes successives de plus en plus complexes, permettant ainsi d'aborder des notions ordonnées de la plus simple à la plus compliquée. Basés sur cette observation, des travaux ont été menés afin d'appliquer ce concept aux algorithmes d'apprentissage automatique (Bengio *et al.*, 2009). Ces travaux ont montré l'intérêt de l'organisation des exemples d'apprentissage d'un même ensemble de données pour l'amélioration de la vitesse de convergence et des performances de généralisation des systèmes entraînés. Il s'agit du curriculum d'apprentissage.

D'autres travaux se sont concentrés sur l'application des principes de transfert d'apprentissage (Weinshall *et al.*, 2018; Pan & Yang, 2010) pour ordonner les exemples appris par le système. Ils ont de nouveau montré l'intérêt d'un curriculum d'apprentissage, notamment pour l'amélioration des performances de généralisation pour des tâches complexes.

Récemment, il a été proposé par Ghannay *et al.* (2018) une première approche de bout en bout permettant de réaliser, de manière totalement disjointe, soit une reconnaissance d'entités nommées, soit une extraction de concepts sémantiques dans la parole. La proposition consistait en l'ajout des frontières des concepts directement dans les séquences à produire par le système. Traditionnellement, les tâches d'extraction de concepts sémantiques et de reconnaissance d'entités nommées dans la parole s'effectuent par l'intermédiaire d'une chaîne de composants. Un système de reconnaissance de la parole constitue le premier composant, puis un système de traitement du langage est appliqué sur les transcriptions automatiques produites par ce premier composant. Le système de traitement du langage est appliqué sur des transcriptions imparfaites. L'avantage d'un système de bout en bout réside dans sa possibilité à limiter la transmission des erreurs de transcription de la parole. Mais aussi, dans l'optimisation jointe des composants de parole et de traitement de la langue pour la tâche finale.

Dans cet article, nous présentons un système qui s'appuie entièrement sur une architecture neuronale similaire au système de reconnaissance de la parole DeepSpeech 2 de Baidu (Amodei *et al.*, 2016). Nous entraînons ce système pour réaliser l'extraction des concepts sémantiques dans les données MEDIA (Devillers *et al.*, 2004). En considérant que la reconnaissance d'entités nommées et l'extraction de concepts sémantiques (comme proposé dans la campagne d'évaluation MEDIA) sont deux tâches proches, nous souhaitons explorer la possibilité d'exploiter des données d'entités nommées pour améliorer notre système d'extraction de concepts. L'application d'un principe de curriculum d'apprentissage est motivée par l'observation du caractère plus générique des annotations d'entités nommées par rapport aux annotations de concepts sémantiques.

Dans nos travaux, nous adaptons le curriculum pour mettre en place une chaîne d'entraînements successifs ordonnés du plus général au plus spécifique. Il s'agit d'une chaîne d'apprentissage pilotée par une stratégie de curriculum. Elle exploite le transfert d'apprentissage, qui nous permet de faire face au manque de données annotées pour la tâche finale. Nous utilisons également un modèle de langage 5-gramme, ainsi qu'un mode étoile permettant de concentrer le système sur les concepts et leurs valeurs. Notre meilleur système obtient des performances bien meilleures que l'état de l'art, qu'il conviendra toutefois de nuancer.

La structure de cet article est la suivante : nous présentons tout d'abord des travaux similaires dans la deuxième section. Puis, dans la troisième section, nous expliquons l'architecture neuronale utilisée. Ensuite, la quatrième section est dédiée à la présentation des ensembles de données et du mode étoile que nous utilisons dans nos travaux. La section suivante présente la chaîne d'apprentissage mise en place. Enfin, l'avant dernière section est consacrée à nos résultats expérimentaux avant de conclure.

2 Travaux similaires

Les récentes avancées des systèmes de reconnaissance de la parole de bout en bout permettent désormais d'obtenir des performances solides (Amodei *et al.*, 2016). Des travaux ont été réalisés sur des tâches appliquées à la parole. Certains se concentrent sur la traduction automatique de bout en bout, faisant suite aux travaux précurseurs de Bérard *et al.* (2016). Ils mettent en place un système neuronal de type encodeur-décodeur, basé sur un mécanisme d'attention. Ce système est entraîné à l'aide d'un petit ensemble de données français parlé - anglais écrit et a la particularité d'utiliser de la parole de synthèse. Les travaux de Weiss *et al.* (2017) sont similaires mais utilisent des données réelles. Ils exploitent un modèle neuronal initialement utilisé en reconnaissance de la parole. Un système du même type est aussi exploité dans les travaux de Bérard *et al.* (2018), dont l'originalité tiens dans l'augmentation de données d'apprentissage par l'utilisation d'un système de traduction texte vers texte. Ces approches sont prometteuses, mais n'ont pas atteint des performances à l'état de l'art, comme le montrent les résultats de la campagne d'évaluation IWSLT 2018 pour la traduction de l'anglais parlé vers l'allemand écrit (Jan *et al.*, 2018).

D'autres travaux concernant la détection d'intention et la détection de domaine ont été réalisés par Serdyuk *et al.* (2018). Ces travaux mettent en œuvre un réseau de neurones inspiré des technologies de reconnaissance de la parole. Ce système obtient d'excellents résultats sur la tâche de détection de domaine, mais n'arrive pas à atteindre des performances à l'état de l'art pour la tâche de détection d'intention. Il montre toutefois l'intérêt de partir directement du signal de parole pour la tâche de compréhension du langage. Des conclusions similaires sont partagées par Ghannay *et al.* (2018) pour la reconnaissance des entités nommées ou l'extraction de concepts sémantiques dans le cadre d'une tâche de *slot filling*.

Enfin, les récents travaux de Platanios *et al.* (2019) ont montré l'intérêt d'une approche exploitant l'apprentissage par curriculum pour une tâche de traduction automatique. Dans ces travaux, les auteurs ont défini les notions de « difficulté » d'un exemple d'apprentissage et de « compétence » d'un modèle appris, dans le cadre de la traduction automatique. Ces deux notions permettent de filtrer les exemples d'apprentissage pour les présenter des plus simples au plus compliqués.

3 Système neuronal

L'architecture neuronale que nous utilisons pour ces travaux est très similaire au système de reconnaissance de la parole DeepSpeech 2 de Baidu (Amodei *et al.*, 2016). Il est composé d'un empilement de couches de convolution (CNN), de couches récurrentes bidirectionnelles (biLSTM), d'une couche de convolution lookahead, d'une couche entièrement connectée et enfin d'une couche softmax. Ce système utilise en entrée des log-spectrogrammes de l'audio calculés sur des fenêtres de 20ms et prédit des séquences de caractères.

Pour nos expérimentations, nous utilisons une implémentation [1] avec deux couches CNN et cinq couches biLSTM. Notre sortie peut correspondre soit à la meilleure hypothèse produite par le système, soit à une hypothèse recalculée à l'aide de l'algorithme « beam search » et d'un modèle de langage 5-gramme. L'architecture neuronale est synthétisée en figure 1.

1. https ://github.com/SeanNaren/deepspeech.pytorch

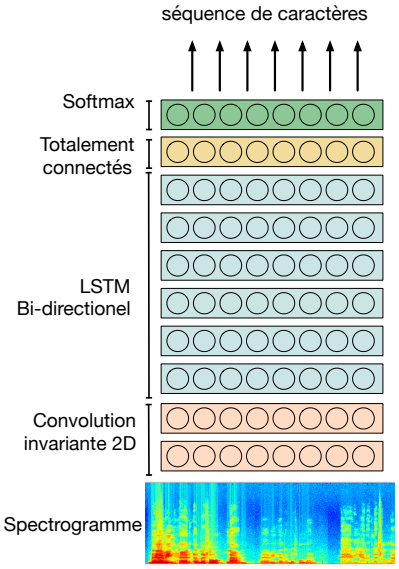

FIGURE 1 – Architecture neuronale DeepSpeech 2

Ce système est entraîné de bout en bout en utilisant la fonction de coût Connectionist Temporal Classification (CTC) (Graves *et al.*, 2006). Une particularité de cette fonction est qu'elle ne nécessite pas d'alignement a priori entre les entrées et les sorties du système. Il est appris pendant l'entraînement.

Pour apprendre cet alignement, les log-spectrogrammes des séquences d'audio sont découpés en tranches. Il s'agit de tranches de taille arbitraire fixe permettant la prédiction d'étiquettes de sorties. Pour chaque tranche, une distribution de probabilités est calculée sur l'ensemble des étiquettes prédictibles (caractères). La séquence produite est constituée de l'ensemble des étiquettes les plus probables de chaque tranche.

Comme la parole humaine est fluctuante en vitesse, un caractère est susceptible d'être représenté par plusieurs tranches. C'est pourquoi, les répétitions sont supprimées dans les séquences produites. Cette réduction empêche l'alignement correct d'un segment audio avec une séquence contenant normalement des répétitions (« curriculum » deviendrait « curiculum »). Pour prendre en compte les répétitions, un caractère spécifique est ajouté dans les étiquettes prédictibles par le système, noté ϵ dans l'exemple de la figure 2. Cette étiquette permet de délimiter deux caractères identiques successifs. Elle ne représente donc aucune information devant être conservée et sera supprimée des séquences finales produites après réduction des répétitions. De plus, le flux de parole n'est pas systématiquement constant sur un segment, il peut être composé de silence. Il est donc inutile de forcer l'alignement de ces portions à une étiquette porteuse de sens. Le caractère spécifique utilisé pour gérer les répétitions est aussi utilisé pour les silences composant l'audio. La figure 2 représente un exemple de l'alignement appris par la fonction CTC. La modélisation de séquence par l'intermédiaire de cette fonction est davantage détaillée par Hannun (2017).

L'alignement entre l'audio et les séquences à produire est appris pendant l'entraînement. Ainsi, l'information relative aux entités nommées et aux concepts sémantiques peut être injectée dans les séquences cibles. Dans nos travaux, nos modifications interviennent au niveau des séquences de caractères présentées au système. Les frontières des concepts y sont ajoutées. Le système produit donc des séquences de caractères au sein desquelles des caractères spécifiques représentent les frontières des concepts.

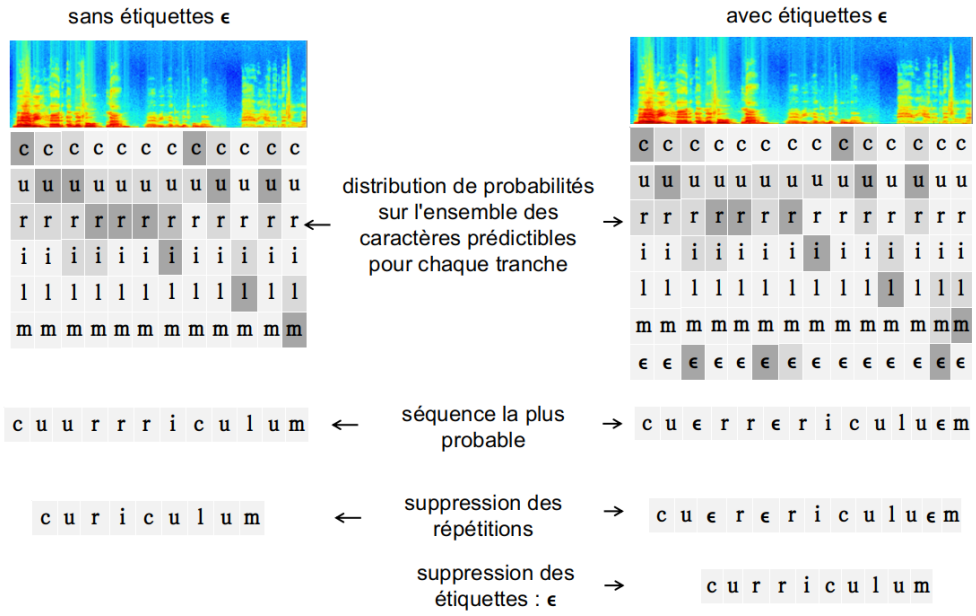

FIGURE 2 – exemple d'alignements de la fonction CTC

4 Données, Augmentation de données, Mode étoile

Dans ces travaux, nous utilisons plusieurs ensembles de données dans le but de maximiser notre quantité de données audio, de données annotées en entités nommées et de données annotées en concepts sémantiques.

4.1 Audio

Parmi les données audio que nous utilisons dans cette étude, une grande partie est issue d'enregistrements d'émissions de radios et de chaînes de télévision francophone. Chaque ensemble est transcrit manuellement et est séparé en trois parties distinctes : développement, test et apprentissage. Les corpus utilisés sont : EPAC (Esteve *et al.*, 2010), ESTER2 (Galliano *et al.*, 2009), ETAPE (Gravier *et al.*, 2012), QUAERO (Grouin *et al.*, 2011), et REPERE (Giraudel *et al.*, 2012).

À ces données, s'ajoutent les ensembles MEDIA (Devillers *et al.*, 2004), PORTMEDIA (Lefèvre *et al.*, 2012) et DECODA (Bechet *et al.*, 2012). Ces données sont composées de conversations téléphoniques enregistrées dans un contexte de réservations d'hôtel, de réservations de billet de spectacle et de conversations d'un centre d'appel de la RATP.

La table 1 illustre la quantité de parole de nos ensembles de données.

Nos ensembles de données proviennent de types d'enregistrements différents. Les données issues des émissions de radio et de télévision ont un enregistrement studio en 16 Khz, tandis que les données téléphoniques ont un enregistrement en 8 Khz. Pour nos travaux, nous avons sous-échantillonné les données studio en 8 Khz pour les additionner aux données téléphoniques.

Pour créer notre corpus audio, nous avons respecté les répartitions officielles de tous nos ensembles de données. Il totalise ainsi 26 heures de développement, 55 heures de test et 357 heures d'entraînement.

Corpus	développement	test	entraînement
EPAC	0	9,5	81
ESTER 2	5	6	89
ETAPE	6,5	6,5	19
QUAERO	0	6,5	68,5
REPERE	3	9,5	35,5
MEDIA	1,5	5	17
PORTMEDIA	2	3,5	7
DECODA	8	8,5	40

TABLE 1 – Répartition des corpus audio utilisés. Les quantités d'audio sont exprimées en heures. Ces valeurs sont calculées sur les segments de parole.

4.2 Entités nommées

Les ensembles de données ETAPE et QUAERO possèdent des transcriptions manuelles annotées en entités nommées. Le formalisme QUAERO (Rosset *et al.*, 2011) est utilisé pour cette annotation. Une entité nommée est définie par un type et une valeur, par exemple la valeur « Paris » possède le type « loc.adm.town ». Avec ce formalisme, les types d'entités nommées possèdent une hiérarchie. En reprenant l'exemple de la valeur « Paris », le type principal « loc » est complété par les sous-types « adm » et « town ». À la hiérarchie s'ajoute une annotation en composant. Ils permettent de décrire davantage les entités nommées. Par exemple « firstname », « lastname » pour une entité de type « pers ». Enfin, les annotions d'entités peuvent être imbriquées. Un exemple selon l'annotation Quaero est : « **<pers.ind** le **<func.ind** président **> <pers.ind <firstname** Emmanuel **> <lastname** Macron **> > >** ».

Pour nos travaux, nous avons simplifié les annotations. Les composants ont été retirés. La hiérarchie des entités nommées est également supprimée. Nous ne conservons que le type principal. Enfin, l'imbrication a été retirée en ne conservant que les annotations les plus proches du niveau mot. Ainsi, nous obtenons huit catégories pour nos annotations : « pers », « func », « org », « loc », « prod », « amount », « time » et « event ». L'exemple précédent devient : « le **<func** président **> <pers** Emmanuel Macron **>** ».

Comme vu dans la section 3, la fonction CTC permet d'apprendre l'alignement entre l'audio et les séquences à produire. Afin d'entraîner notre système, nous ajoutons les frontières des entités nommées dans les séquences de caractères. Ainsi, la séquence servant à apprendre un système de reconnaissance de la parole « le sculpteur césar est mort hier à paris » devient « le sculpteur **<pers** césar **>** est mort **<time** hier **>** à **<loc** paris **>** ». Comme le système utilisé dans cette étude produit des séquences de caractères, les informations de frontières d'entités nommées sont représentés par un caractère. Nous ajoutons dans les étiquettes prédictibles par la fonction CTC huit caractères pour les balises ouvrantes et un caractère pour la balise fermante.

Les ensembles QUAERO et ETAPE représentent en totalité 107 heures d'audio transcrites et annotées manuellement. Pour maximiser notre quantité de données audio annotées en entités nommées, nous avons utilisé le système NeuroNLP2 [2]. Nous entraînons un modèle à l'aide des données manuelles modifiées pour respecter notre annotation simplifiée. Puis nous l'appliquons aux transcriptions manuelles des données audio, d'émission de radio et de TV, n'ayant pas d'annotation en entités nommées. Notre

2. https ://github.com/XuezheMax/NeuroNLP2

augmentation automatique porte sur les corpus EPAC, REPERE, ESTER 2. Nous composons notre corpus audio annoté en entités nommées avec les annotations manuelles et automatiques. Il représente 14,5 heures de développement, 44 heures de test et 293 heures d'apprentissage.

4.3 Concepts sémantiques

L'ensemble de données cible de l'extraction de concepts sémantiques est MEDIA. Il est composé de 1 257 dialogues séparés en trois parties. Une partie développement comprenant 1 300 phrases, une partie test comprenant 3 500 phrases et une partie apprentissage comprenant 17 700 phrases. Cet ensemble de données est annoté selon 76 concepts sémantiques (Bonneau-Maynard *et al.*, 2005). Ces concepts sont par exemple : « chambre-type », « hotel-etat », « sejour-nbNuit », « temps-jour-semaine ». Par observation, un lien peut être fait entre les annotations en concepts sémantiques et les annotations en entités nommées. Il est aisé de rapprocher les concepts « amount » et « nombre-reservation », ou encore de rapprocher « loc » et « localisation-ville ». Nous avons pu observer que les entités nommées « pers », « loc », « amount », « time » et « event » peuvent être reliées aux annotations en concepts sémantiques du corpus MEDIA. Nous considérons que les concepts définis dans ce corpus sont plus précis que les entités nommées.

Nous avons augmenté notre quantité de données annotées en concepts sémantiques en utilisant le corpus PORTMEDIA. Il est composé de 700 dialogues et contient 10 400 phrases. Cet ensemble de données a été produit dans le but d'étudier la portabilité de domaine. Ainsi, il est proche de l'ensemble MEDIA. Il est annoté selon 36 concepts sémantiques, par exemple « nb-billets » ou « type-billet ». Il y a 26 concepts en commun avec l'annotation de MEDIA (« command-tache », « paiement-montant-entier », ...). Les concepts de PORTMEDIA peuvent aussi être rapprochés des concepts d'entités nommées (qui sont toujours considérés plus génériques).

Dans nos travaux, nous exploitons les données de concepts sémantiques de la même manière que les données d'entités nommées. C'est-à-dire en injectant les frontières des concepts directement dans les séquences que le système devra produire.

4.4 Mode étoile

Dans les travaux de Ghannay *et al.* (2018), il a été proposé un mode étoile ayant pour but d'aider le système à se concentrer sur les concepts et leurs valeurs. La mise en œuvre de ce mode consiste à remplacer l'ensemble des caractères en dehors des concepts et de leurs valeurs par une étoile « * ». Ce remplacement s'effectue dans les séquences cibles utilisées pour l'apprentissage du modèle. Ainsi, l'exemple de la section 4.2 devient « * **<pers** césar **>** * **<time** hier **>** * **<loc** paris **>** ».

La fonction de coût CTC donne la même importance à chaque caractère émis composant une séquence. Les éléments de contexte ont donc une importance réduite vis-à-vis des concepts et de leurs valeurs. Ils sont représentés par un seul caractère « * » contre trois caractères ou plus pour les concepts et leurs valeurs (un caractère pour l'ouverture de concept, au minimum un caractère pour la valeur, et un caractère pour la fermeture du concept). Avec cette réduction de l'ensemble du contexte, une erreur faite sur les concepts et valeurs sera nécessairement plus pénalisante pour le système, ce qui lui permettra de concentrer son apprentissage.

5 Chaîne d'apprentissage

Le transfert d'apprentissage rend possible l'exploitation de données issues d'un espace de caractéristiques proches, afin d'apprendre des connaissances a priori facilitant l'entraînement sur une tâche finale (Pan & Yang, 2010). Il est possible d'apprendre un système à partir d'un système existant, plutôt qu'en repartant de zéro. Cette approche a un intérêt dans le cadre d'un manque de données dans un domaine cible. Pour notre étude, l'ensemble d'apprentissage de MEDIA est composé de seulement 17h de parole. Le transfert d'apprentissage permet d'exploiter l'intégralité des données audio à notre disposition pour pré-apprendre un système de reconnaissance de la parole. Ce système pourra ensuite être spécialisé avec les données MEDIA.

L'apprentissage par curriculum repose sur l'introduction ordonnée des différents concepts à apprendre au sein d'un même ensemble. Ce qui permet à un système d'exploiter les concepts plus généraux pour apprendre les concepts plus spécifiques (Bengio *et al.*, 2009). Un système peut converger davantage et plus rapidement en exploitant les exemples d'apprentissage d'un même ensemble du plus simple au plus compliqué. Dans le cas de nos données, les liens entre les entités nommées et les concepts sémantiques nous permettent de considérer l'ordre de complexité suivant : I) Les données audio sont les plus simples. II) Les données annotées en entités nommées sont plus complexes. III) Enfin, les données annotées en concepts sémantiques sont les plus complexes puisque ces concepts sont plus précis que les entités nommées.

Nos travaux ne s'intègrent pas entièrement dans une démarche de curriculum d'apprentissage, puisque nos données ne sont pas issues d'un même ensemble. Ils ont pour objectifs de tirer parti à la fois du transfert d'apprentissage et du curriculum d'apprentissage. Nous réalisons donc un système appris par transfert d'apprentissage et piloté par une stratégie de curriculum. La mise en œuvre d'un tel système nous permet de compenser notre manque de données pour la tâche finale et d'optimiser davantage le modèle obtenu. Nous réalisons une chaîne d'apprentissages successifs dans laquelle nous organisons les corpus de données utilisés du plus générique au plus spécifique.

Entre chaque étape d'apprentissage, nous conservons les poids de l'ensemble du modèle obtenu comme initialisation du réseau pour l'étape suivante. Toutefois, nous réinitialisons totalement la couche haute (softmax) afin de prédire de nouvelles étiquettes. À chaque étape, les étiquettes de sorties du système sont modifiées en fonction des données utilisées.

Dans nos travaux, nous distinguons quatre étapes :

— ASR : Entraînement d'un modèle de reconnaissance de la parole.
— NER : Entraînement d'un modèle de reconnaissance des entités nommées.
— PM+M : Entraînement d'un modèle d'extraction de concepts sémantiques.
— M : Optimisation d'un modèle sur l'ensemble de données MEDIA.

Lors de l'étape « ASR », nous utilisons la totalité des données présentées dans la section 4.1. L'étape « NER » quant à elle est réalisée à l'aide des données augmentées de la section 4.2. Enfin, l'étape « PM+M » correspond à un apprentissage sur l'ensemble augmenté des données décrites dans la section 4.3.

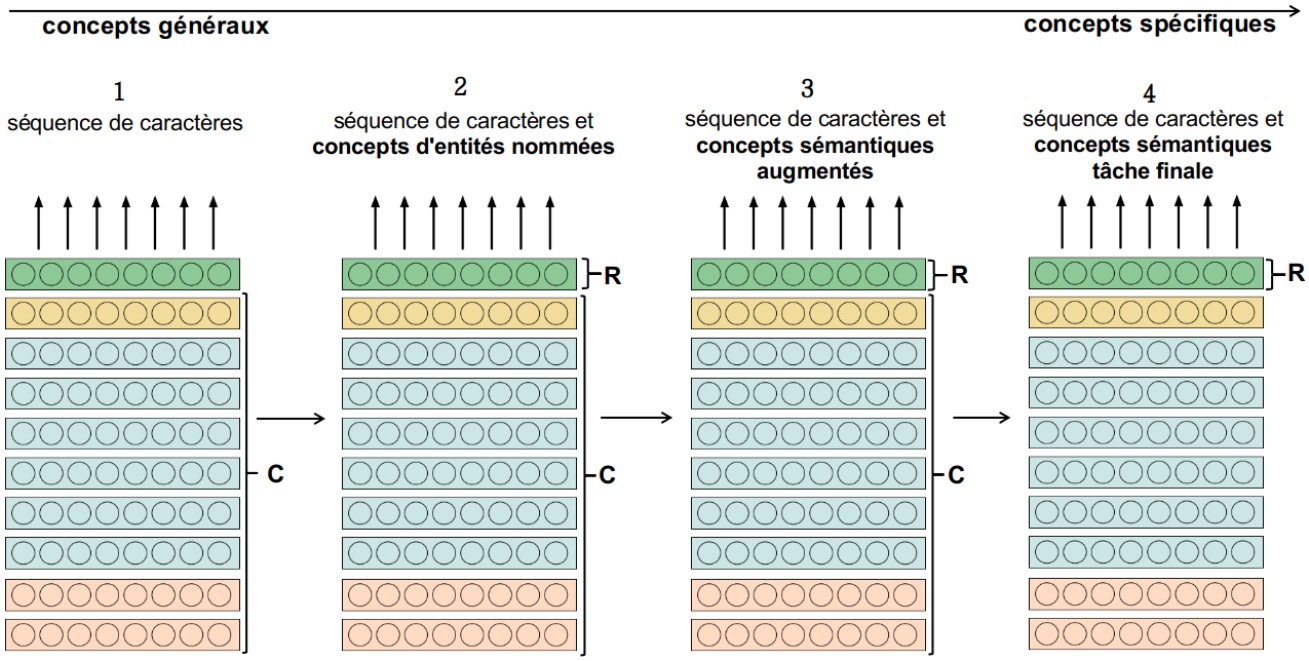

FIGURE 3 – Représentation de la chaîne d'apprentissage mise en place. « C » signifie que les poids sont conservés, « R » signifie qu'ils sont réinitialisés.

6 Expérimentations

Dans cette section, nous présentons les résultats obtenus par nos systèmes sur l'ensemble de données de test de MEDIA. Nos expérimentations sont évaluées selon deux métriques. Le Concept Error Rate (CER), qui consiste en la somme des erreurs de substitution, d'insertion et de suppression divisée par le nombre de références. C'est une métrique proche du Word Error Rate (WER). Cependant, elle permet d'évaluer uniquement la reconnaissance des types de concepts dans les séquences produites, par exemple « chambre-type » / « command-tache ». La seconde métrique utilisée est le Concept Value Error Rate (CVER) qui se calcule comme le CER, mais qui évalue à la fois les types de concepts et leurs valeurs. C'est-à-dire les mots associés aux types de concepts.

L'état de l'art concernant l'extraction de concepts sémantiques au sein de l'ensemble MEDIA se situe à 19,9 % de CER et 25,1 % de CVER (Simonnet *et al.*, 2017). Le système utilisé est une chaîne de composants. Tout d'abord un système de reconnaissance de la parole, puis un système de traitement du langage. Ce dernier exploite les transcriptions automatiques du premier composant. La qualité des transcriptions automatique joue un rôle important dans la qualité finale du système. Les performances du système de transcription de la parole utilisé par l'état de l'art étaient de 23,6 % de WER.

Plusieurs expérimentations contrastives ont été menées afin de vérifier l'impact de chacune des étapes d'apprentissage. Nous avons aussi mené des expérimentations avec des maillons de la chaîne en mode étoile (noté *).

6.1 Mode purement neuronal

chaîne d'apprentissage	CER	CVER
M	39,8	52,1
ASR $\rightarrow$ *M*	23,7	30,3
ASR $\rightarrow$ *NER* $\rightarrow$ *M*	22,4	28,7
ASR $\rightarrow$ *PM+M* $\rightarrow$ *M*	22,2	28,8
ASR $\rightarrow$ *NER* $\rightarrow$ *PM+M* $\rightarrow$ *M*	21,6	27,7
ASR $\rightarrow$ *NER* $\rightarrow$ *PM+M* $\rightarrow$ *M**	**20,1**	27,2
ASR $\rightarrow$ *NER* $\rightarrow$ *PM+M** $\rightarrow$ *M**	**20,1**	**26,9**

TABLE 2 – Résultats expérimentaux exprimés en CER et CVER.

Les résultats expérimentaux montrent l'intérêt des étapes d'apprentissages successives. La table 2 présente les résultats selon la première hypothèse de sortie du système.

L'ensemble de données MEDIA seul n'est clairement pas suffisant pour atteindre de bonnes performances. L'apprentissage d'un système de reconnaissance de la parole comme première étape avant un entraînement sur MEDIA, $ASR \rightarrow M$, permet une amélioration conséquente des performances du système. Les résultats montrent une descente du CER de 39,8 % (M) à 23,7 % ($ASR \rightarrow M$). Cette première étape sert de socle commun à l'ensemble des expérimentations suivantes.

L'utilisation des données d'entités nommées dans la chaîne $ASR \rightarrow NER \rightarrow M$ montre une utilité en raison d'une amélioration significative de 1,3 point de CER, par rapport à la chaîne $ASR \rightarrow M$. Nous atteignons un CER de 22,4 %, soit un gain relatif de 5,5 %. L'exploitation de l'augmentation des données MEDIA avec l'ensemble PORTMEDIA nous permet aussi un gain en descendant de 23,7 % à 22,2 % de CER (comparaison des chaînes $ASR \rightarrow M$ et $ASR \rightarrow PM + M \rightarrow M$). Enfin, nous avons appris un modèle exploitant pleinement le transfert d'apprentissage piloté par la stratégie de curriculum proposé dans ce papier. Ce modèle, $ASR \rightarrow NER \rightarrow PM + M \rightarrow M$, atteint 21,6 % de CER. Il montre à nouveau l'utilité des données d'entités nommées, par une amélioration de 0,6 point. Soit un gain relatif de 2,7 %.

L'utilisation du mode étoile, décrit dans la section 4.4, aide le système à ce concentrer sur les concepts et leurs valeurs. Ce qui nous permet d'améliorer une dernière fois le CER de 1,5 %. Nous atteignons une valeur de 20,1 % (CER).

Toutes les observations que nous avons réalisées sur les taux de CER peuvent être effectuées sur les taux de CVER.

6.2 Mode à double étape avec modèle de langage n-gramme

Le système deepspeech 2 permet de recalculer ses sorties avec un modèle de langage. Le modèle de langage utilisé est au niveau mot. Il est appris à l'aide de l'ensemble d'apprentissage de MEDIA, ainsi que d'un ensemble conséquent de données issues d'articles de journaux. Les informations de concepts sémantiques, encodées sur un caractère, sont conservées pour l'entraînement du modèle de langage. Un second est appris avec le mode étoile. Lors de nos expérimentations, nous avons calculé des modèles variant de 3-gramme à 6-gramme. Nous utilisons les modèles 5-gramme, puisque nous n'observons plus d'amélioration significative au-delà. Les résultats de nos expérimentations avec les modèles de langage sont reportés dans la table 3.

chaîne d'apprentissage	CER	CVER
M	32,8	37,9
ASR $\rightarrow$ *M*	20,1	24,0
ASR $\rightarrow$ *NER* $\rightarrow$ *M*	18,8	22,8
ASR $\rightarrow$ *PM+M* $\rightarrow$ *M*	19,0	22,9
ASR $\rightarrow$ *NER* $\rightarrow$ *PM+M* $\rightarrow$ *M*	18,1	22,1
ASR $\rightarrow$ *NER* $\rightarrow$ *PM+M* $\rightarrow$ *M**	16,6	21,3
ASR $\rightarrow$ *NER* $\rightarrow$ *PM+M** $\rightarrow$ *M**	**16,4**	**20,9**

TABLE 3 – Résultats expérimentaux exprimés en CER et CVER. Les résultats sont recalculés à l'aide d'un modèle de langage 5-gramme.

L'utilisation des modèles de langage nous permet une amélioration significative des résultats sur l'ensemble de nos expériences. L'intérêt des entités nommées dans notre chaîne d'apprentissage est conservé. En comparant les systèmes $ASR \rightarrow M$ et $ASR \rightarrow NER \rightarrow M$, nous observons un gain relatif de 6,5 % de CER et en comparant les systèmes $ASR \rightarrow PM+M \rightarrow M$ et $ASR \rightarrow NER \rightarrow PM+M \rightarrow M$, il est de 4,7 %.

Par l'utilisation du mode étoile, nous obtenons désormais nos meilleurs taux de CER et de CVER, avec respectivement 16,4 % et 20,9 %. Pour obtenir ces résultats, nous avons utilisé le mode étoile pendant les étapes d'apprentissage manipulant les concepts sémantiques. L'application du mode étoile aux données d'entités nommées a pour effet de dégrader nos résultats. Nous observons que le mode étoile doit être appliqué au niveau des étapes d'extraction de concepts sémantiques. En comparant nos meilleurs résultats à l'état de l'art (CER : 19,9 % et CVER : 25,1 %), nous obtenons de très bons résultats. Nous avons un gain relatif de 17,6 % pour le CER, et de 16,7 % pour le CVER.

Le système que nous avons mis en œuvre obtient un WER de 10,1 %. Il est calculé sur les sorties de la chaîne d'apprentissage $ASR \rightarrow NER \rightarrow PM+M \rightarrow M$. Ces sorties sont recalculées avec le modèle de langage 5-gramme, puis toutes les informations de concepts sémantiques sont filtrées pour évaluer le WER sur les mots uniquement.

Nous obtenons dans notre approche un WER bien meilleur que l'état de l'art, 10,1 % contre 23,6 % (soit 57,2 % de gain relatif). La comparaison de nos résultats n'est pas entièrement juste et il conviendrait de mettre en œuvre le système à chaîne de composants de l'état de l'art avec des transcriptions automatiques d'une qualité similaire à notre système.

6.3 Validation de la stratégie de curriculum

Pour compléter nos expérimentations, nous avons tenté l'apprentissage de la chaîne $ASR \rightarrow PM+M \rightarrow NER \rightarrow M$, qui rompt le processus itératif de spécialisation des différentes étapes d'apprentissage en inversant les étapes NER et $PM+M$. Nous observons lors de l'apprentissage de l'étape NER que le modèle ne converge pas, et ce malgré nos différentes tentatives de modification des paramètres d'apprentissage, notamment le taux d'apprentissage. Cette absence de convergence confirme l'apport de l'ordonnancement des étapes d'apprentissage que nous avons présenté dans notre stratégie de curriculum. Ainsi, en ayant inversé les étapes NER et $PM+M$, nous ne pouvons pas tirer parti des connaissances portées par les données étiquetées en entités nommées.

7 Conclusion

Ce travail présente l'intérêt de l'utilisation des données d'entités nommées pour la reconnaissance de concepts sémantiques à travers la mise en place d'une chaîne d'apprentissages successifs pilotée par une stratégie de curriculum. Nous avons réalisé des augmentations de données via un ensemble proche (PORTMEDIA) pour les concepts sémantiques et via une augmentation artificielle (NeuroNLP2) pour les entités nommées. Nos données viennent alimenter le système DeepSpeech 2. Pour ce faire, nous avons intégré les frontières des entités nommées et des concepts sémantiques directement dans les chaînes de caractères à produire. Les données sont présentées au système de manière à apprendre une tâche de reconnaissance de la parole, puis une tâche de reconnaissance des entités nommées dans la parole et enfin une tâche d'extraction de concepts sémantiques.

Les résultats expérimentaux montrent une amélioration des performances de cette approche, qui permet un gain relatif allant de 2,7 % à 6,5 % de Concept Error Rate. Notre meilleur système est obtenu en exploitant pleinement la chaîne d'apprentissage pilotée par la stratégie de curriculum proposée dans ce papier. Ce système utilise également un mode étoile, lui permettant de se concentrer sur les concepts et leurs valeurs, ainsi qu'un modèle de langage 5-gramme pour recalculer ses sorties. Il surpasse l'état de l'art sur la tâche d'extraction de concepts sémantiques dans MEDIA, avec un gain relatif de 17,6 % de Concept Error Rate.

Il convient cependant de nuancer nos résultats, puisque notre système obtient de bien meilleures performances en termes de reconnaissance de la parole que le composant parole du système état de l'art. Nos expérimentations ont montré un gain relatif de 57,2% de WER. Un complément nécessaire à ce travail consiste en l'utilisation de transcriptions automatiques ayant des performances similaires à notre approche avec le système état de l'art.

En conclusion, ces travaux présentent des premiers résultats intéressants à propos de l'intégration des entités nommées dans une chaîne d'apprentissage, basée sur les principes de transfert et de curriculum d'apprentissage, pour l'extraction de concepts sémantiques. Ils constituent un socle intéressant pour de futurs travaux visant à approfondir les possibilités de l'exploitation des entités nommées comme données génériques, notamment pour la portabilité de domaine. Ce socle peut également être amélioré par l'ajout d'informations supplémentaires pouvant aider l'extraction de concepts sémantiques. En effet, nos expérimentations se basent uniquement sur les mots et les concepts pour effectuer la tâche finale. Nous pouvons envisager de profiter des années de travaux effectuer dans le domaine du traitement du langage pour enrichir nos données d'apprentissage, comme par exemple l'utilisation d'étiqueteurs morpho-syntaxiques ou sémantique.

Remerciements

Ce travail a été soutenu par le RFI Atlanstic2020 à travers le projet RAPACE (Réseaux de neurones profonds pour le traitement de la langue orale et écrite). Il a également été soutenu par l'agence ANR au travers du projet CHIST-ERA ON-TRAC, sous le numéro de contrat : ANR-18-CE23-0021-01. Les auteurs souhaitent remercier Sean Naren pour la mise à disposition de son implémentation du système DeepSpeech 2, ainsi que Xuezhe Ma pour son implémentation du système NeuroNlp2.

Références

AMODEI D., ANANTHANARAYANAN S., ANUBHAI R., BAI J., BATTENBERG E., CASE C., CASPER J., CATANZARO B., CHENG Q., CHEN G. *et al.* (2016). Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of the thirty-third International Conference on Machine Learning (ICML'16)*, p. 173–182, New York, United States.

BECHET F., MAZA B., BIGOUROUX N., BAZILLON T., EL-BEZE M., DE MORI R. & ARBILLOT E. (2012). Decoda : a call-centre human-human spoken conversation corpus. In *Proceedings of the eighth Language Resources and Evaluation Conference (LREC'12)*, p. 1343–1347, Istanbul, Turkey.

BENGIO Y., LOURADOUR J., COLLOBERT R. & WESTON J. (2009). Curriculum learning. In *Proceedings of the twenty-sixth International Conference on Machine Learning (ICML'09)*, p. 41–48, Montreal, Canada.

BÉRARD A., BESACIER L., KOCABIYIKOGLU A. C. & PIETQUIN O. (2018). End-to-end automatic speech translation of audiobooks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18)*, p. 6224–6228, Calgary, Canada.

BÉRARD A., PIETQUIN O., BESACIER L. & SERVAN C. (2016). Listen and translate : A proof of concept for end-to-end speech-to-text translation. In *NIPS Workshop on End-to-end Learning for Speech and Audio Processing*, Barcelona, Spain.

BONNEAU-MAYNARD H., ROSSET S., AYACHE C., KUHN A. & MOSTEFA D. (2005). Semantic annotation of the french media dialog corpus. In *Proceedings of the ninth European Conference on Speech Communication and Technology (EUROSPEECH'05)*, p. 3456–3459, Lisbon, Portugal.

DEVILLERS L., MAYNARD H., ROSSET S., PAROUBEK P., MCTAIT K., MOSTEFA D., CHOUKRI K., CHARNAY L., BOUSQUET C., VIGOUROUX N. *et al.* (2004). The french media/evalda project : the evaluation of the understanding capability of spoken language dialogue systems. In *Proceedings of the fourth Language Resources and Evaluation Conference (LREC'04)*, p. 2131–2134, Lisbon, Portugal.

ESTEVE Y., BAZILLON T., ANTOINE J.-Y., BÉCHET F. & FARINAS J. (2010). The epac corpus : Manual and automatic annotations of conversational speech in french broadcast news. In *Proceedings of the seventh international conference on Language Resources and Evaluation (LREC'10)*, p. 1686–1689, Malta.

GALLIANO S., GRAVIER G. & CHAUBARD L. (2009). The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In *Proceedings of the tenth Conference of the International Speech Communication Association (INTERSPEECH'09)*, p. 2543–2546, Brighton, United Kingdom.

GHANNAY S., CAUBRIÈRE A., ESTÈVE Y., CAMELIN N., SIMONNET E., LAURENT A. & MORIN E. (2018). End-to-end named entity and semantic concept extraction from speech. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'19)*, p. 692–699, Athens, Greece.

GIRAUDEL A., CARRÉ M., MAPELLI V., KAHN J., GALIBERT O. & QUINTARD L. (2012). The repere corpus : a multimodal corpus for person recognition. In *Proceedings of the eighth Language Resources and Evaluation Conference (LREC'12)*, p. 1102–1107, Istanbul, Turkey.

GRAVES A., FERNÁNDEZ S., GOMEZ F. & SCHMIDHUBER J. (2006). Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the twenty-third International Conference on Machine Learning (ICML'06)*, p. 369–376, Pittsburgh, United States.

GRAVIER G., ADDA G., PAULSON N., CARRÉ M., GIRAUDEL A. & GALIBERT O. (2012). The etape corpus for the evaluation of speech-based tv content processing in the french language. In *Proceedings of the eighth Language Resources and Evaluation Conference (LREC'12)*, p. 114–118, Istanbul, Turkey.

GROUIN C., ROSSET S., ZWEIGENBAUM P., FORT K., GALIBERT O. & QUINTARD L. (2011). Proposal for an extension of traditional named entities : From guidelines to evaluation, an overview. In *Proceedings of the fifth Linguistic Annotation Workshop*, p. 92–100, Portland, United States.

HANNUN A. (2017). Sequence modeling with ctc. *Distill*. https ://distill.pub/2017/ctc.

JAN N., CATTONI R., SEBASTIAN S., CETTOLO M., TURCHI M. & FEDERICO M. (2018). The iwslt 2018 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, p. 2–6, Bruges, Belgium.

LEFÈVRE F., MOSTEFA D., BESACIER L., ESTÈVE Y., QUIGNARD M., CAMELIN N., FAVRE B., JABAIAN B. & ROJAS-BARAHONA L. (2012). Robustesse et portabilités multilingue et multi-domaines des systèmes de compréhension de la parole : les corpus du projet portmedia. In *Proceedings of the Joint Conference JEP-TALN-RECITAL*, p. 779–786, Grenoble, France.

PAN S. J. & YANG Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, p. 1345–1359.

PLATANIOS E. A., STRETCU O., NEUBIG G., POCZOS B. & MITCHELL T. M. (2019). Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv :1903.09848*.

ROSSET S., GROUIN C. & ZWEIGENBAUM P. (2011). *Entités nommées structurées : guide d'annotation Quaero*. LIMSI-Centre national de la recherche scientifique.

SERDYUK D., WANG Y., FUEGEN C., KUMAR A., LIU B. & BENGIO Y. (2018). Towards end-to-end spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18)*, p. 5754–5758, Calgary, Canada.

SIMONNET E., GHANNAY S., CAMELIN N., ESTÈVE Y. & DE MORI R. (2017). ASR error management for improving spoken language understanding. In *Proceedings of the eight-teenth Conference of the International Speech Communication Association (INTERSPEECH'17)*, p. 3329–3333, Stockholm, Sweden.

WEINSHALL D., COHEN G. & AMIR D. (2018). Curriculum learning by transfer learning : Theory and experiments with deep networks. *arXiv preprint arXiv :1802.03796*.

WEISS R. J., CHOROWSKI J., JAITLY N., WU Y. & CHEN Z. (2017). Sequence-to-sequence models can directly translate foreign speech. In *Proceedings of the eight-teenth Conference of the International Speech Communication Association (INTERSPEECH'17)*, p. 2625–2629, Stockholm, Sweden.

Détection des ellipses dans des corpus de sous-titres en anglais

Anissa Hamza[1] Delphine Bernhard[1]
(1) LiLPa - EA 1339, Université de Strasbourg, France
hamzaa@unistra.fr, dbernhard@unistra.fr

RÉSUMÉ

Cet article présente une méthodologie de détection des ellipses en anglais qui repose sur des patrons combinant des informations sur les tokens, leur étiquette morphosyntaxique et leur lemme. Les patrons sont évalués sur deux corpus de sous-titres. Ces travaux constituent une étape préalable à une étude contrastive et multi-genres de l'ellipse.

ABSTRACT

Ellipsis Detection in English Subtitles Corpora

This article presents a methodology for detecting ellipses in English based on patterns combining information on tokens, their part-of-speech tags and their lemma. The patterns are evaluated on two subtitles corpora. This work is a preliminary step towards a contrastive and multi-genre study of the ellipsis phenomenon.

MOTS-CLÉS : ellipse, anglais, corpus, sous-titres, détection automatique.

KEYWORDS: ellipsis, English, corpus, subtitles, automated detection.

1 Introduction

L'ellipse renvoie à une incomplétude syntaxique de la phrase qui ne présente pas d'incidence sur la transmission de son contenu sémantique. En effet, grâce à la présence d'un antécédent linguistique ou extralinguistique, le co-locuteur parvient à interpréter la forme invisible d'une séquence donnée et à établir la relation avec le reste des éléments dans le discours. De cette définition, découlent deux principes fondamentaux pour que l'ellipse soit fonctionnelle : (i) une structure syntaxique incomplète `[s]` créant ainsi un vide syntaxique dans la structure de la phrase (en d'autres termes, ce vide ne contient aucune forme linguistique) ; (ii) un antécédent `[a]` qui permet la récupération et la transmission du contenu sémantique des éléments effacés. Une relation de dépendance entre le site elliptique et l'antécédent est ainsi apparente. L'exemple 1 ci-dessous [1] illustre un cas de *pseudogapping* représenté par l'ensemble vide ∅, où le syntagme verbal *put the first one through you* est ellipsé pour éviter la répétition et alléger la forme de la phrase. Ainsi, le syntagme effacé *put the first one through you* est nécessaire uniquement pour la construction syntaxique de la phrase puisque son sens est déduit du contexte, ici, linguistique [2].

(1). Get down ! Get down off it, you old cuckold, I don't care who you are. I'll **put the first one through you !**$_{[a]}$ I swear it, I **will** ∅$_{[s]}$ now ! One ! Two !

(2). Descends ! Descends de là, vieux connard ! Je me fous que ce soit toi. Je tire, je te tire dessus ! Je te jure que **je vais tirer** maintenant. Un ! Deux !

1. Les sites elliptiques sont marqués par ∅ dans les exemples.
2. Sauf mention contraire, les exemples sont extraits du corpus de développement décrit dans la section 3.1.

D'un point de vue contrastif [3], cette ellipse n'est pas autorisée en français puisque les auxiliaires des temps composés ne peuvent apparaître seuls dans la phrase. Comme le montre l'exemple ci-dessus, le site elliptique est traduit par la périphrase *aller+inf* qui exprime le futur proche *je vais tirer*. L'ellipse post-auxiliaire est ainsi rendue impossible **je vais ∅* puisque *aller* est, en quelque sorte un pseudo-auxiliaire qui ne peut apparaître seul dans cette configuration.

Si certaines catégories de l'ellipse semblent simples à repérer et à caractériser, d'autres présentent davantage de défi à relever tant au niveau théorique qu'au niveau appliqué. L'objectif de cette contribution est ainsi de décrire une méthodologie de détection globale du phénomène elliptique en anglais, mettant en avant les différents enjeux qu'il pose aux outils du TAL, notamment aux étiqueteurs morphosyntaxiques. L'identification automatique des ellipses constitue la première étape d'un travail visant une analyse contrastive (anglais-français) et multi-genres du phénomène elliptique, afin de comprendre son fonctionnement en discours et les problèmes posés pour la traduction, et notamment la traduction automatique.

Nous présentons dans un premier temps une typologie des ellipses en anglais, en vue de leur détection automatique (section 2). Nous décrivons ensuite les corpus utilisés et les patrons d'identification définis par rapport à cette typologie (section 3). Enfin, nous détaillons une évaluation de ces patrons (section 4).

2 Typologie des ellipses à détecter en anglais

Les études théoriques de l'ellipse ont permis d'établir un éventail de classifications. Ces classifications varient d'une approche à l'autre : classification établie selon la composition syntaxique et l'agencement grammatical, selon la situation pragmatique, ou alors selon le contexte proprement dit. Nous suivons ici la taxonomie des ellipses établie par les approches syntaxiques contemporaines de l'étude de l'ellipse, notamment celle de van Craenenbroeck & Merchant (2013). Sachant que certaines ellipses ne figurent dans aucune classification (ellipse du sujet et de l'auxiliaire) et compte tenu des problèmes qu'elles posent lors d'une traduction automatique, nous avons néanmoins décidé de procéder à leur détection. van Craenenbroeck & Merchant retiennent dans leur classification trois types : les ellipses du syntagme verbal dans lesquelles sont classées les ellipses verbales et les ellipses post-auxiliaires, les ellipses propositionnelles dans lesquelles sont classés le *sluicing* et ses sous-catégories, et enfin les ellipses nominales. Afin d'atteindre notre objectif de détection automatique, nous avons simplifié la catégorisation de ces ellipses en les classant uniquement par élément déclencheur et en n'établissant pas de sous-catégorie. Ainsi, nous avons élaboré notre classification en la fondant entièrement sur les critères morphosyntaxiques qu'il est possible de formaliser dans les outils dont nous disposons. À titre indicatif, on trouvera ci-dessous des exemples illustrant les catégories qui seront détectées automatiquement. Les noms utilisés pour les patrons de détection présentés dans la section 3 sont indiqués entre parenthèses.

Ellipse du syntagme verbal

— L'ellipse verbale est l'omission du syntagme verbal et de ses compléments, laissant visible uniquement l'auxiliaire ou l'opérateur. Dans l'exemple ci-dessous, l'ellipse est déclenchée par l'opérateur *do* (`post-do`).
(3). John plays the piano but Maria doesn't ∅.

3. Approcher l'ellipse d'un point de vue contrastif sert à révéler davantage d'irrégularités pouvant contribuer à comprendre sa nature et à la définir. En effet, si ce phénomène passe généralement inaperçu au sein d'une même langue, sa complexité est vite mise en avant, grâce à un effet miroir, lors du passage à une autre langue, puisque certaines langues ne l'autorisent que rarement.

— L'ellipse post-auxiliaire renvoie à l'omission du groupe verbal déclenchée soit par un modal soit par un auxiliaire :
  — Déclenchée par un modal (`post-mod`)
    (4). Lauren can play the guitar and Mike can ∅, too. (Merchant, 2019)
  — Déclenchée par une inversion sujet-verbe ou une *question tag* (`vs-tag`)
    (5). Sit down. Should I ∅ ?
  — Déclenchée par *have* ou *be* (`post-be/have`)
    (6). Are you leaving tomorrow ? No, I'm not ∅".
  — *Pseudogapping*. En anglais, le *pseudogapping* est une ellipse déclenchée par un auxiliaire (`post-aux`), un opérateur (`post-do`) ou un modal (`post-mod`) : la forme non finie du verbe (cas des modaux et de *do*), ou le participe (cas des auxiliaires), sont omis, laissant après l'auxiliaire ou le modal une partie du prédicat.
    (7). John invited Sarah, and Mary did ∅ Jane. (Gengel, 2013)
  — Déclenchée par le marqueur de l'infinitif *to* (`post-to`)
    (8). Do you want me to ∅ ?

Ellipse propositionnelle

— *Gapping* : ellipse où le verbe fini est effacé dans une ou plusieurs constructions parallèles ou propositions coordonnées :
  (9). Mary carries a suitcase, and John ∅ a bag.
— *Sluice* : omission de la proposition entière à l'exception du pronom *wh-* comme dans l'exemple 10 (`post-wh`). Selon le nombre et le type des éléments restant après le pronom *wh-*, une sous-catégorie du *sluice* peut être identifiée [4].
  (10). Someone is knocking at the door, but I don't know who ∅.
— Ellipse dans les questions fragmentaires (`qs-frag`) : cette ellipse a été identifiée lors de l'analyse de certains types de discours (dialogue informels notamment). Elle renvoie à l'omission de l'auxiliaire (dans les temps composés) ou de l'opérateur (dans les temps simples) accompagné du sujet dans les interrogatives, laissant visibles dans la phrase seulement le verbe et ses compléments [5].
  (11). ∅ Going somewhere ?
  (12). ∅ Eat something ? No.

Ellipse nominale Compte tenu de sa rareté dans plusieurs langues, cette ellipse est marginalement étudiée. La liste ci-dessous n'est pas exhaustive mais représente les ellipses que nous avons sélectionnées pour la détection automatique, notamment en raison des problèmes qu'elles posent à la traduction automatique (par exemple de l'anglais vers l'arabe) :

— Ellipse déclenchée par le *'s* du génitif (`post-geni`)
  (13). He took John's car but not Mary's ∅.
— Ellipse déclenchée par un quantifieur (`post-quant`)
  (14). Thank you, but I already have some ∅.
— Ellipse déclenchée par un nombre (`post-card` et `post-ord`)
  (15). If they have eggs, bring me six ∅.
  (16). I have two interesting books, the first ∅ is in French while the second ∅ is in English.

4. Dans la présente contribution, aucune distinction n'est faite entre les sous-catégories du *sluice*.
5. On peut aussi trouver des questions fragmentaires où l'auxiliaire est ellipsé comme dans *You married ?* Nous n'envisageons pas le traitement de ces occurrences ici.

3 Corpus et méthodologie

3.1 Corpus de développement et d'évaluation

Les catégories d'ellipses décrites dans la section précédente sont détectées à l'aide de patrons. Les patrons de détection ont été mis au point manuellement à partir d'un corpus de développement de 5 362 tokens regroupant 331 exemples d'ellipses. Ces occurrences, repérées manuellement dans leur contexte, et mêlées à 120 phrases non-elliptiques, sont toutes extraites de documents authentiques publiés entre 1960 et 2014 et qui n'ont aucun lien avec le corpus d'évaluation : pièces de théâtre (H. Pinter), nouvelles (G. Green, F. Forsyth, J. Arden, ...), articles de presse (The Guardian, Mail), dialogues et romans (M. Barbery, J. Coe). Pour compléter ce corpus, nous avons également utilisé des exemples issus de (McShane & Babkin, 2016) et (Rønning *et al.*, 2018b) afin d'augmenter le nombre d'occurences elliptiques et couvrir ainsi le plus de variation possible.

Afin d'évaluer la performance des patrons, deux sous-corpus d'évaluation de tailles différentes ont été utilisés. Ils appartiennent tous deux au registre conversationnel de sous-titres. En effet, l'ellipse, en tant que propriété de discours spontané, est plus fréquente dans ce type de discours (Baird *et al.*, 2018). Par ailleurs, le corpus des sous-titres sélectionné propose également une version française, ce qui permettra à l'avenir de réaliser une étude contrastive. Le premier (Corpus 1) est extrait de séries répertoriées dans Opus *OpenSubtitles* [6] (Tiedemann, 2012), et compte 197 302 tokens et 1 270 occurrences d'ellipses. Le deuxième (Corpus 2) contenant 36 676 tokens et 396 occurrences d'ellipses, est constitué des sous-titres de séries Broadchuch et Downton Abbey, récupérés des DVD (Strong & Lyn, 2018; Percival *et al.*, 2018) et compilés par nous-même. Pour repérer les types d'ellipses à l'aide des patrons établis, les deux corpus, de développement et d'évaluation, ont été étiquetés morphosyntaxiquement à l'aide de l'étiqueteur morphosyntaxique de Stanford (Manning *et al.*, 2014). Par ailleurs, ces corpus ont été annotés manuellement par l'une des auteurs en ajoutant un code selon l'élément déclencheur de l'ellipse. Une vérification a ensuite été effectuée par les deux auteurs, à l'aide d'expressions régulières simples afin de détecter les occurrences éventuellement manquantes non annotées. En effet, l'ellipse reste un phénomène peu fréquent et il est donc nécessaire de parcourir une grande quantité de texte pour obtenir un nombre satisfaisant d'occurrences, ce qui augmente la possibilité d'oublier des occurrences lors de l'annotation.

3.2 Définition de patrons de détection

La détection des ellipses repose sur des patrons définis à l'aide de l'outil *TokensRegex*, inclus dans Stanford CoreNLP (Chang & Manning, 2014), qui permet de rechercher des séquences de tokens en combinant différentes informations (forme, lemme, partie du discours, etc.). Nous avons préféré des patrons opérant sur les tokens plutôt que sur l'analyse syntaxique en dépendance car ces derniers sont difficiles à mettre en oeuvre dans certains cas, qui nécessitent de prendre en compte l'ordre des mots dans la phrase (ellipses déclenchées dans les *question tags* ou par l'inversion du sujet et du verbe). Nous avons aussi observé de nombreuses erreurs dans les analyses syntaxiques obtenues.

La Table 1 donne quelques exemples de patrons (la totalité des patrons définis pour chaque type d'ellipse est présentée dans la Table 3). Un ou plusieurs patrons ont été établis pour identifier une même catégorie d'ellipse, en prenant en compte les conditions morpho-syntaxiques de chaque catégorie présentée en section 2. Les patrons sont relativement longs et détaillent toutes les conditions possibles de façon à atteindre le plus grand degré de précision possible. Par exemple, certains comportements syntaxiques récurrents sont observables dans le cas de l'ellipse du sujet et de l'auxiliaire dans la

6. `http://opus.nlpl.eu/OpenSubtitles-v2016.php`

question fragmentaire (`qs-frag`, voir exemples 11 et 12). En effet, si la phrase commence par un verbe principal (étiqueté *VB*) et se termine par un point d'interrogation, on peut supposer qu'il y a deux vides syntaxiques : un vide sujet et un vide auxiliaire. Le type de l'auxiliaire omis est ensuite interprété selon la flexion du verbe :

— S'il est réduit à un participe passé *ed* ou présent *ing*, l'auxiliaire absent peut être soit *have* soit *be*,

— S'il est une forme non-finie, l'opérateur *do* ou un modal manque,

L'un des patrons définis pour les questions fragmentaires est présenté dans la Table 1. Il repère une phrase contenant un verbe non-conjugué et se terminant par un point d'interrogation. Seuls trois types de tokens peuvent précéder ce verbe, de manière optionnelle : un tiret ou un point (marqueurs typographiques d'un dialogue dans les sous-titres), éventuellement suivis d'un adverbe ou d'un adjectif, puis d'une virgule. Par ailleurs, on ne devra pas trouver un pronom personnel après le verbe.

Type	**Patron**
`qs-frag`	`^/[-.]+/? [{pos:/RB|JJ/}]? /,/? [{pos:/VB|VB[^PZD].*/}] [!{pos:/PRP/} & ! /,/] []* /[?]/`
`post-do`	`[!{pos:/W.*/}][{pos:/(PRP.?|NN.?)/}] []{0,2} [{lemma:do} & !{pos:/VB[GN]/}] [{lemma:not}]? [{pos:PRP}]? /[.;!:?,]+/`
`post-geni`	`[!{lemma:of}] [!{pos:PRP}] [pos:POS] /[.;!?:,]/`

TABLE 1: Exemples de patrons de détection.

Pour l'ellipse `post-do`, plusieurs critères syntaxiques sont à considérer pour que *do* soit déclencheur d'une ellipse. Il est déclencheur lorsque dans la phrase il est un :

— verbe de suppléance :
(17). They sang a song or mother **did** ∅.

— auxiliaire de négation :
(18). Do you believe in miracles ? **I don't** ∅.

— auxiliaire d'inversion contrainte par certains adverbes :
(19). He didn't allow him to speak. Neither **did** I ∅.

— auxiliaire emphatique dans les assertives affirmatives qui ne contiennent pas d'auxiliaire (*have/be*) ou de modal :
(20). You just missed a toast. Oh yes, you **did** ∅.

Ces quatre cas autorisent le vide syntaxique laissé par l'omission du verbe lexical qui, s'il était restitué, n'impacterait pas la grammaticalité de la phrase, à condition que l'ensemble du syntagme verbal (VB et compléments) soit restitué. De plus, *do* est supplétif dans la mesure où il ne se charge pas *entièrement* du sens du verbe de la phrase, mais le *complète*. Il est ainsi considéré comme déclencheur d'une ellipse post-auxiliaire. Le patron `post-do` présenté dans la Table 1 récupérera donc toute phrase contenant le lemme *do* précédé par un pronom personnel ou un nom et non précédé par un pronom *wh-*. Le lemme *do* peut être suivi par *not* marqueur de négation ou d'un pronom personnel mais ne devra pas être suivi par une forme verbale.

Un dernier exemple que nous pouvons illustrer ici est le cas du génitif `post-geni`. Dans les trois possibilités qui existent pour exprimer la possession en anglais (la préposition *of*, les pronoms possessifs *his, hers, yours, etc.* et le morphème *–'s/–s'*), seul le morphème *–s* est considéré comme déclencheur d'une ellipse nominale comme dans :

(21). But Rosetti was writing about her own death, not her grandparents' ∅.

En effet, l'utilisation des pronoms possessifs est strictement anaphorique et n'engendre aucun vide syntaxique dans la structure de la phrase. Le patron récupérera ainsi tous les tokens étiquetés POS (possession), non précédés par un pronom personnel ou par le lemme *of*[7], suivis immédiatement d'un signe de ponctuation.

3.3 Limites de la méthodologie

De la liste présentée dans la section 2, seul le *gapping* ne fera pas l'objet d'une détection automatique. En effet, en l'absence d'un élément déclencheur, il est difficile d'établir avec notre méthodologie un patron qui prendrait en compte toutes les variations et les propriétés syntaxiques d'une occurrence de *gapping* et ce quel que soit le niveau de l'analyse, syntaxique ou morphosyntaxique.

Pour ce qui est de l'analyse morphosyntaxique, l'établissement d'un patron pour détecter ce type d'ellipse est particulièrement complexe, compte tenu du nombre variable des éléments résiduels dans la phrase et de la multiplicité de leurs étiquettes morphosyntaxiques. L'absence du verbe entre le sujet et son complément d'objet, par exemple, peut difficilement être formalisée dans un patron TokensRegex, en raison des faux positifs que le patron pourrait repérer. En effet, la figure 1 ci-dessous illustre deux occurrences étiquetées : la première est une phrase non-elliptique et présente le cas de deux éléments (*a candy* et *a cake*) coordonnés avec *and* et étiquetés `NN`. Ces deux éléments n'entretiennent pas de relation sujet-objet comme c'est le cas de la deuxième phrase où *and* coordonne deux propositions, dans une configuration très similaire du point de vue de la séquence des étiquettes morphosyntaxiques (`DT NN CC DT NN`). Le verbe de la deuxième proposition (*the chief ∅ a bag*) est omis présentant de ce fait un *gapping*.

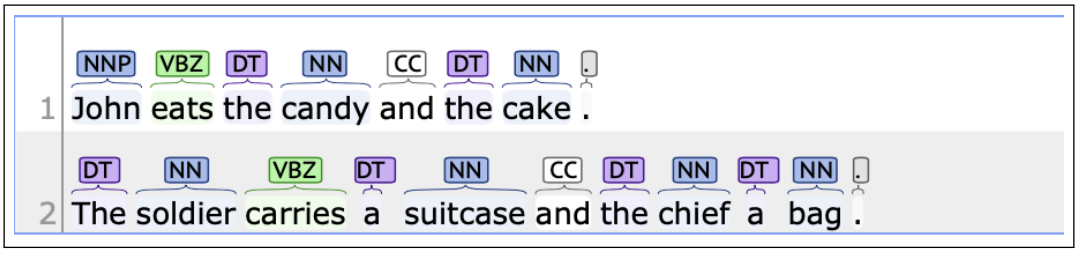

FIGURE 1: Exemple d'étiquetage d'une phrase non-elliptique et du *gapping*.

La détection du *stripping*, sous-catégorie du gapping, aurait pu être envisagée grâce aux marqueurs *too, as well*, et *also* toujours présents après le site elliptique, ou aux conjonctions de coordination *and* et *or* qui le précèdent :

(22). Jane Likes apples and Maria ∅ too.

Cependant, ce type de cas est très marginal et n'est pas représentatif du *gapping*. De la même manière les autres configurations, notamment les constructions parallèles déclenchant ce phénomène de *gapping*, ne pourront pas être détectées.

De ce fait, compte tenu des limites de TokensRegex d'une part, et de la complexité à fixer des conditions stables du phénomène elliptique dans le *gapping* d'autre part, nous avons dû écarter sa détection automatique[8].

7. Les constructions *friend of, part of* ne sont pas traitées comme elliptiques ici.
8. D'autres types d'ellipse que nous n'avons pas mentionnés dans la classification sont également difficiles à détecter automatiquement tels que *Bare ellipsis Argument* et les réponses fragmentaires.

4 Évaluation et discussion des résultats

Les résultats obtenus pour chaque type d'ellipse dans les trois corpus utilisés sont détaillés dans la Table 2, selon l'élément déclencheur.

Le discours conversationnel dont les sous-titres font partie est représentatif d'une communication caractérisée par une parole spontanée [9] et interactionnelle, susceptible d'offrir un usage de l'ellipse très varié. On remarque par exemple que les ellipses déclenchées par les modaux, *be/have, to, do, wh*, et les *questions tag* `vs-tag`, toutes apparentées à l'ellipse du syntagme verbal et propositionnelle sont beaucoup plus fréquentes que l'ellipse nominale. L'échange ci-dessous contenant deux ellipses, une déclenchée par *can* et l'autre par *to*, partageant le même antécédent *speak to Mark*, est repéré par deux patrons `post-mod` et `post-to` :

(23). - Beth, have you spoken to Mark ? - I can't ∅. - You've got to ∅.

Les patrons sont bien adaptés aux types d'ellipses qui sont présents dans le corpus de développement (sauf pour `qs-frag`). Il reste difficile d'y inclure toutes les variations syntaxiques qui peuvent se trouver dans un corpus de grande taille, comme le montrent les performances moindres obtenues sur les corpus d'évaluation 1 et 2. À l'exception de 4 catégories (`post-card, post-quant, post-be/have, vs-tag`), les résultats montrent une F-mesure plus élevée dans le corpus 2 que dans le corpus 1. Le rappel est globalement élevé, ce qui répond à nos besoins pour une étude ultérieure des ellipses dans des corpus de grande taille. Il est en effet plus simple de filtrer des faux positifs manuellement que de parcourir exhaustivement des corpus de grande taille pour retrouver les faux négatifs. Par ailleurs, même si le corpus de développement intègre des exemples issus de différents genres textuels, il faudra encore tester les patrons sur d'autres genres (discours parlementaire, journalistique, littérature, textes scientifiques, ...). Enfin, il restera à améliorer certains patrons : `qs-frag`, `post-be/have`, `post-geni`. Les résultats plus faibles observés pour ces types d'ellipse peuvent être justifiés par leur rareté et leur complexité qui se manifeste par des configurations non-observées dans un corpus de développement de petite taille.

Type d'ellipse	**Corpus de dév.**				**Corpus 1**				**Corpus 2**			
	#	P	R	F1	#	P	R	F1	#	P	R	F1
`post-do`	34	0,97	1,00	0,99	188	0,57	0,88	0,69	45	0,69	0,93	0,79
`post-mod`	72	0,96	0,96	0,96	147	0,71	0,96	0,81	44	0,79	0,86	0,83
`vs-tag`	31	0,96	0,87	0,92	305	0,58	0,97	0,72	72	0,58	0,93	0,72
`post-be/have`	31	0,84	0,84	0,84	185	0,47	0,70	0,57	52	0,50	0,67	0,57
`post-to`	30	1,00	0,93	0,97	39	0,62	0,87	0,72	36	0,79	0,83	0,81
`post-wh`	43	0,93	1,00	0,97	223	0,44	0,98	0,61	82	0,62	0,99	0,76
`qs-frag`	32	0,94	0,53	0,68	90	0,40	0,74	0,52	41	0,73	0,54	0,62
`post-card`	6	1,00	1,00	1,00	67	0,28	0,94	0,43	10	0,21	0,70	0,33
`post-geni`	19	1,00	0,89	0,94	5	0,25	0,60	0,35	12	0,83	0,83	0,83
`post-quant`	29	1,00	0,93	0,96	17	1,00	0,94	0,97	1	0,50	1,00	0,67
`post-ord`	4	1,00	0,75	0,86	4	0,50	1,00	0,67	1	1,00	1,00	1,00

TABLE 2: Résultats de l'évaluation. La colonne # correspond au nombre d'instances.

Les patrons n'ont pas toujours atteint une précision parfaite lors de leur application sur le corpus d'évaluation, notamment la catégorie `post-card` avec une précision de 0,28 dans le corpus 1 et

9. Les acteurs suivent un script imitant les conversations spontanées.

de 0,21 dans le corpus 2, ou encore la catégorie de post-geni qui a atteint seulement 0,25 dans le corpus 1. Après avoir parcouru les ellipses détectées ou non par chaque patron, les problèmes semblent être liés à deux types d'erreurs :

— Erreurs dues à la précision insuffisante des patrons qui renvoient :
  — au manque d'étiquettes suffisamment représentatives et précises pour annoter les déclencheurs cruciaux de l'ellipse par exemple la non-distinction entre *to* préposition *went to+nom* et *to* marqueur d'infinitif *want to+verb* (24) ou entre *do* comme auxiliaire suppletif/opérateur et *do* comme verbe plein (25) :
  (24). a. All_DT my_PRP$ life_NN I_PRP 've_VBP wanted_VBD somebody_NN to_TO talk_VB to_TO._. (Non elliptique)
  b. You_PRP can_MD call_VB me_PRP a_DT sap_VBP if_IN you_PRP want_VBP to_TO ∅._.
  (25). a. What_WP are_VBP you_PRP trying_VBG to_TO do_VB,—, Babe_NNP ?_. (Non elliptique)
  b. I_PRP know_VBP him_PRP better_JJR than_IN they_PRP do_VBP ∅._.
  — à la difficulté d'affiner le patron pour prendre en compte davantage de variations des structures elliptiques. On trouve par exemple des cas d'ellipse qs-frag dans les déclaratives, qui ont été annotées manuellement comme elliptiques, mais non détectés car aucun patron n'a été dédié aux phrases déclaratives. On pourrait en effet obtenir beaucoup de faux-positifs détectés dans les phrases à l'impératif :
  (26). Read_VB that_DT book_NN ?_. (qs-frag ellipse d'un pronom et de *have*)
  (27). Read_VB that_DT book_NN now_RB._. (impératif non elliptique)
  (28). Read_VB that_DT book_NN ._. (ellipse d'un pronom personnel et de *have* dans la réponse fragmentaire)

— Erreurs engendrées par l'étiqueteur : ces erreurs sont liées au mauvais étiquetage de certaines catégories, en raison de l'ambiguïté qu'un seul mot peut présenter. En effet, le *'s* comme marqueur du génitif *customer's* et le *'s* de la forme contractée de *be* ou *have* à la 3e personne *father's been there* (29) sont tous les deux étiquetés comme POS ci-dessous :
  (29). a. Whose car is this ? The_DT costumer_NN 's_POS ∅._.
  b. Father_NN 's_POS been_VBN there_RB._. (Non elliptique)
  Par ailleurs, les deux phrases ci-dessous sont détectées comme ellipses qs-frag pourtant la première ne l'est pas.
  (30). Can_MD you_PRP open_VB up_RP,_. please_ VB ?_.
  (31). We_PRP was_VBD hired_VBN as_IN his_PRP$ bodyguard_NN,_. see_VB ?_.
  Dédié à repérer toute occurrence contenant un verbe non précédé d'un nom et suivi d'un ? le patron a détecté *please*, ici incorrectement étiqueté VB au lieu de UH (interjection) comme elliptique. Cette ambiguïté morphosyntaxique est également à l'origine de la non détection des ellipses par le patron comme dans l'échange *Flattering her ? Oh yes he is* où *flattering* est étiqueté NN au lieu de VBG.

Par le biais de ces erreurs nous pouvons pointer les difficultés et les limites des patrons à base de tokens dues à un étiquetage pas suffisamment détaillé, voire erroné, ou encore en raison de l'instabilité du phénomène elliptique. Malgré ces restrictions, l'utilisation de patrons à l'aide de tokens reste toutefois, pour l'instant, une opération avantageuse dans la mesure où elle offre une méthodologie simple de détection. Elle pourrait être améliorée si d'autres paramètres étaient inclus dans l'élaboration des patrons. On pourrait par exemple enrichir le corpus de développement. Nous nous sommes limitées dans cet article à 331 occurrences elliptiques pour développer nos patrons mais une collecte manuelle d'occurrences diverses et variées et en aussi grand nombre que possible, pour fastidieuse qu'elle soit,

pourrait contribuer au perfectionnement des patrons. De plus, dans la mesure où *do, have, be* et *to* sont des déclencheurs essentiels de l'ellipse et compte tenu d'absence d'étiquettes les distinguant des autres catégories, il serait utile de parvenir à les étiqueter de manière plus fine en distinguant leur utilisation en tant d'auxiliaires et verbes pleins.

5 État de l'art

Les études de l'ellipse en TAL notamment pour l'anglais, restent peu nombreuses par rapport aux investigations théoriques. Le nombre limité de ces recherches est indubitablement lié aux défis que présente le phénomène elliptique. Comment la détecter quand on la remarque à peine ? Bos & Spenader (2011) par exemple expliquent le manque de recherches effectuées sur la détection automatique des ellipses, notamment des ellipses verbales, par la présence de deux difficultés principales. La première tient à l'absence d'outils ou de procédures servant à localiser l'ellipse et son antécédent, et la seconde, aux lacunes des recherches théoriques qui, une fois l'ellipse et son antécédent approximativement repérés, se concentrent sur la tâche de résolution, négligeant alors le problème de l'antécédent. À la lumière des travaux théoriques existants, il apparaît que les trois éléments-clefs à prendre en compte dans une étude de l'ellipse sont le site elliptique, l'antécédent et le contexte (syntaxique et/ou sémantique).

Pour ce qui est du site elliptique, il est important et nécessaire de définir ce qui manque (la nature et la fonction des éléments ellipsés). Le site elliptique semble poser des problèmes au TAL dans la mesure où le nombre de tokens ellipsés varie d'une occurrence à une autre. Pour ce qui est de l'antécédent, il est important de le localiser si la résolution de l'ellipse est envisagée au même titre que sa détection. Ceci peut indéniablement aider à délimiter le site elliptique. En revanche, le problème qui se pose, pour les outils du TAL en particulier, concerne la nature même de l'antécédent. Que faire dans le cas d'un antécédent extralinguistique ? Enfin le contexte : toutes les ellipses ne peuvent pas être détectées via la syntaxe. De plus, même dans le cas où la syntaxe semble suffire, le même type d'ellipse peut révéler des variations dépendantes de paramètres externes à la syntaxe. En effet, le genre et le type de discours peuvent affecter les comportements syntaxiques de l'ellipse.

Compte tenu de ces éléments clefs, les travaux sur l'analyse automatique de l'ellipse distinguent plusieurs étapes : (1) la détection des sites elliptiques [10] (2), la détection et la délimitation de l'antécédent, (3) la résolution de l'ellipse (comblement du site elliptique pour reconstituer une phrase non-elliptique). Pour ce qui est des types d'ellipses traitées, les ellipses du syntagme verbal restent les plus étudiées en traitement automatique des langues. Il existe également quelques travaux traitant du *sluicing* ou *sluice*, de l'ellipse du sujet ou du *gapping*.

McShane & Babkin (2016) décrivent un système, appelé ViPER (*VP Ellipsis Resolver*) qui vise à détecter et résoudre certains types particuliers d'ellipses du syntagme verbal avec un haut niveau de précision. La détection des ellipses traitées se fait en 3 étapes, combinant des patrons de surface et une analyse syntaxique en dépendances avec Stanford CoreNLP. Liu *et al.* (2016) proposent un système par apprentissage pour détecter les ellipses du syntagme verbal, entraîné à partir de deux corpus annotés (celui de Bos & Spenader (2011) et celui de Nielsen (2005)). L'objectif du système est de prendre une décision binaire concernant les modaux et les auxiliaires (site elliptique ou non). Les descripteurs utilisés pour la classification reposent sur l'étiquetage morphosyntaxique, la lemmatisation et l'analyse syntaxique en dépendance. La F-mesure obtenue est 69,52 pour le corpus de Bos & Spenader (2011) et 75,39 pour celui de Nielsen (2005).

10. Appelée détection de la cible (*target detection*) par Liu *et al.* (2016).

De nombreux travaux se focalisent de fait plutôt sur la détection de l'antécédent ou la résolution de l'ellipse, considérant ainsi la première étape de détection des ellipses comme résolue. Hardt (1992) propose un algorithme à base de règles pour déterminer les antécédents des ellipses du syntagme verbal (*Verb Phrase ellipsis*). L'algorithme prend pour entrée un syntagme verbal elliptique et une liste de syntagmes verbaux apparaissant à proximité (même phrase ou deux phrases précédentes). L'algorithme élimine alors les antécédents impossibles et trie les autres syntagmes verbaux par ordre de préférence. Pour les besoins de l'évaluation, les exemples d'ellipses ont été collectés dans le corpus Brown (étiqueté en parties du discours) à l'aide d'expressions régulières détectant les auxiliaires qui n'ont pas de verbe dans le contexte proche. Pour ce qui est du *sluice*, on pourra retenir les travaux récents de Baird *et al.* (2018) et Rønning *et al.* (2018a,b) : le premier article concerne la classification manuelle du type de *sluice* dans un corpus de sous-titres (à partir de *sluices* détectés automatiquement à l'aide d'expressions régulières), tandis que les deux derniers se focalisent sur la résolution des *sluices*. Le *gapping* est traité par Schuster *et al.* (2018) afin de produire des représentations de l'analyse syntaxique en dépendance qui encodent explicitement le matériel élidé. Les expériences présentées reposent notamment sur des phrases sélectionnées à partir d'une relation de dépendance spécifique (`orphan`) dans un des corpus *Universal Dependencies* (UD) pour l'anglais et des phrases collectées manuellement à partir de diverses ressources. Le *gapping* est également au cœur des travaux de Droganova *et al.* (2018a) qui mettent en avant la représentation choisie pour ce phénomène dans les corpus UD, consistant à promouvoir un des dépendants orphelins à la position du parent manquant et conduisant les analyseurs à prédire des relations entre des mots qui ne sont généralement pas reliés par une relation de dépendance. Par ailleurs, le phénomène est rare et donc peu représenté dans les corpus d'entraînement, ce qui complique encore la tâche pour les analyseurs syntaxiques. Pour combler le manque d'exemples dans les corpus d'entraînement, Droganova *et al.* (2018b) ont produit semi-automatiquement des phrases artificielles qui sont similaires à des constructions elliptiques du point de vue de leur structure. Enfin, Droganova & Zeman (2017) mettent en avant les nombreuses erreurs d'annotation manuelle relevées pour les constructions elliptiques dans les corpus UD, ce qui rend compte de la complexité du phénomène.

Les travaux présentés ici se concentrent généralement sur un type particulier d'ellipse, alors que nous proposons de détecter une grande variété de phénomènes elliptiques. Par ailleurs, les objectifs de l'étude de l'ellipse en TAL concernent avant tous les problèmes posés à l'analyse syntaxique (annotation manuelle, analyse automatique). Notre objectif est différent, dans la mesure où nous envisageons une analyse linguistique contrastive (comparaison anglais-français) et multi-genres afin d'étudier l'impact des phénomènes elliptiques sur la traduction humaine et automatique.

6 Conclusion et perspectives

Nous avons présenté une méthodologie de repérage automatique des ellipses en anglais évaluée sur des corpus de sous-titres. Il est important d'ajouter qu'à ce stade, établir des patrons à base de séquences de tokens associés à leur étiquette morphosyntaxique et leur lemme peut paraître rédhibitoire dans le sens où les conditions sont limitées aux tokens dont la moindre variation (même une erreur de tokénisation) entrave la reconnaissance de l'ellipse. Cette étape est cependant un préalable nécessaire à un premier repérage des occurrences elliptiques. L'utilisation de ces patrons peut également constituer une réelle amorce en vue d'une détection et d'une classification des ellipses à base d'un apprentissage automatique. De plus, une détection automatique de l'ellipse et de son antécédent pourrait, à notre sens, contribuer à une classification ainsi qu'à une reconnaissance automatique des genres de discours. En effet, si l'ellipse se révèle comme une particularité d'un genre spécifique, sa détection automatique permettrait certainement de définir le genre de texte analysé. De la même façon, sachant que l'ellipse

reste l'une des erreurs les plus fréquentes et cruciales rencontrées dans la traduction automatique aujourd'hui, aux côtés de la métaphore et certaines figures de style, sa détection et sa résolution informatisées sont ainsi les enjeux inévitables dans la recherche dans ce domaine.

7 Annexe

Type	**Nb de patrons**	**Patrons**
post-do	2	^[{lemma:no}]? [/[.;!:?,]/]? [{pos:/(PRP.?|NN.?)/}] []* [{lemma:do} & !{pos:/VB[GN]/}] [{lemma:not}]? [{pos:PRP}]? /[.;!:?,]+/
		[!{pos:/W.*/}][{pos:/(PRP.?|NN.?)/}] []{0,2} [{lemma:do}& !{pos:/VB[GN]/}] [{lemma:not}]? [{pos:PRP}]? /[.;!:?,]+/
vs-tag	4	[]* [{lemma:/(be|do|have)/} | {pos:MD}] [{lemma:not}]? [{pos:PRP}] /[.!?:]+/
		[]* [{lemma:/(be|do|have)/} | {pos:MD}] [{lemma:not}]? [{pos:DT}]? [{pos:PRP}] /[.]+/$
		^ [{lemma:/(be|do|have)/} | {pos:MD}] [{lemma:not}]? [{pos:/PRP|NN.*/}]? /,/? [{pos:/PRP|NN.*/}]? /[?]+/$
		[]* /,/ [{lemma:/(be|do|have)/} | {pos:MD}] [{lemma:not}]? [{pos:/PRP|NN.*/}]? /,/
post-mod	2	[{pos:/PRP|NN.*|WP|CC/} | {lemma:/,/}] [{pos:RB}]? [{pos:MD}] [{pos:/[^V].*/}]* /[.,:!;?-]+/
		[{pos:/PRP|NN.*|WP/}] [{pos:RB}]? [{pos:MD}] [{pos:CC} | {pos:"IN";lemma:/(if|unless)/}] []* /[.,:!;?-]+/
post-be/have	2	[!{pos:/V.*/} & !{pos:/WP.*/}] [{pos:/PRP.?|NN.?/}] [{pos:RB}]? [{lemma:/be|have/}] [{lemma:not}]? /[.;!?:,-]+/
		^[{pos:/PRP.?|NN.?/}] [{pos:RB}]? [{lemma:/be|have/}] [{lemma:not}]? /[.;!?:,-]+/

TABLE 3: Totalité des patrons utilisés pour la détection. Les patrons sur fond grisé sont expliqués de manière détaillée dans la section 3.2.

Type	Nb de patrons	Patrons
post-to	2	[{pos:/VB.*/ }] [{pos:PRP}]? [{lemma:not}]? [{pos:to}] /[.?,:!;]+/
		[{pos:/VB.*/ }] [{pos:RB}]? [{pos:JJ}]? [{lemma:not}]? [{pos:to}] /[.?,:!;]+/
post-wh	1	[{pos:/^W.*/} & {lemma:/(wh(at|y|o|en|ere|ich|ose)|how)/}] [!{pos:/V.*/} & !{pos:MD}]* [{lemma:/[,.!;:?-]+/} | {pos:CC}]
qs-frag	3	^/[-.]+/? [{pos:/RB|JJ/}]? /,/? [{pos:/VB|VB[^PZD].*/}] [!{pos:/PRP/} & ! /,/] []* /[?]/
		^/[-.]+/? [{pos:/RB|JJ/}]? /,/? [{pos:/VB|VB[^PZD].*/} !{lemma:thank}] [{pos:/PRP|NN/}]? /[?]/
		/,/ [{pos:/VB|VB[^PZD].*/} & !{lemma:thank}] [{pos:/PRP| NN/}]? /[?]/
post-geni	1	[!{lemma:of}] [!{pos:PRP}] [pos:POS] /[.;!?:,]/
post-quant	1	"/[sS]ome|[aA]ny/ /[.;!?:,]+/
post-card	2	[!{pos:/N.*/}] [{pos:CD} & !{lemma:one}] /[.;!?:,]+/
		[!{pos:/N.*/} & !{pos:JJ} & !{lemma:/th(at|ese|is|ose)/}] [{pos:CD} & {lemma:one}] /[.;!?:,]+/
post-ord	1	/the/ [{pos:/JJ.*/;ner:ORDINAL}] /[.;!?:,]+/

TABLE 3: Totalité des patrons utilisés pour la détection (suite).

Références

BAIRD A., HAMZA A. & HARDT D. (2018). Classifying Sluice Occurrences in Dialogue. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, p. 1580–1583.

BOS J. & SPENADER J. (2011). An annotated corpus for the analysis of VP ellipsis. *Language Resources and Evaluation*, **45**(4), 463–494.

CHANG A. X. & MANNING C. D. (2014). *TokensRegex : Defining cascaded regular expressions over tokens*. Rapport interne CSTR 2014-02, Department of Computer Science, Stanford University.

DROGANOVA K., GINTER F., KANERVA J. & ZEMAN D. (2018a). Mind the Gap : Data Enrichment in Dependency Parsing of Elliptical Constructions. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, p. 47–54.

DROGANOVA K. & ZEMAN D. (2017). Elliptic Constructions : Spotting Patterns in UD Treebanks. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, p. 48–57.

DROGANOVA K., ZEMAN D., KANERVA J. & GINTER F. (2018b). Parse Me if You Can : Artificial Treebanks for Parsing Experiments on Elliptical Constructions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, p. 1845–1852.

GENGEL K. (2013). *Pseudogapping and ellipsis*. Oxford, Royaume-Uni : Oxford University Press.

HARDT D. (1992). An algorithm for VP ellipsis. In *Proceedings of the 30th annual meeting of the Association for Computational Linguistics*, p. 9–14.

LIU Z., PELLICER E. G. & GILLICK D. (2016). Exploring the steps of verb phrase ellipsis. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, p. 32–40.

MANNING C. D., SURDEANU M., BAUER J., FINKEL J., BETHARD S. J. & MCCLOSKY D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, p. 55–60.

MCSHANE M. & BABKIN P. (2016). Detection and Resolution of Verb Phrase Ellipsis. *LiLT (Linguistic Issues in Language Technology)*, **13**.

MERCHANT J. (2019). Ellipsis : a survey of analytical approaches. In J. VAN CRAENENBROECK & T. TEMMERMAN, Eds., *The Oxford Handbook of Ellipsis*, p. 19–45. Oxford, Royaume-Uni : Oxford University Press.

NIELSEN L. A. (2005). *A corpus-based study of Verb Phrase Ellipsis Identification and Resolution*. Thèse de doctorat, King's College London.

PERCIVAL B., BOLT B., HALL E., SPIRO M. & ENGLER M. (2018). Downton Abbey - Saisons 1 à 6 - L'intégrale de la série. DVD.

RØNNING O., HARDT D. & SØGAARD A. (2018a). Linguistic Representations in Multi-task Neural Networks for Ellipsis Resolution. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 66–73.

RØNNING O., HARDT D. & SØGAARD A. (2018b). Sluice Resolution without Hand-Crafted Features over Brittle Syntax Trees. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 236–241, New Orleans, Louisiana.

SCHUSTER S., NIVRE J. & MANNING C. D. (2018). Sentences with Gapping : Parsing and Reconstructing Elided Predicates. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL 2018)*.

STRONG J. & LYN E. (2018). Broadchurch Saisons 1 + 2. DVD.

TIEDEMANN J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, p. 2214–2218, Istanbul, Turkey.

VAN CRAENENBROECK J. & MERCHANT J. (2013). *Ellipsis phenomena*, In M. DEN DIKKEN, Ed., *The Cambridge Handbook of Generative Syntax*, p. 701–745. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

La génération automatique de poésie en français

Tim Van de Cruys
IRIT & CNRS
118 Route de Narbonne
31062 Toulouse Cedex 9
France
tim.vandecruys@irit.fr

RÉSUMÉ

La génération automatique de poésie est une tâche ardue pour un système informatique. Pour qu'un poème ait du sens, il est important de prendre en compte à la fois des aspects linguistiques et littéraires. Ces dernières années, un certain nombre d'approches fructueuses sont apparues, capables de modéliser de manière adéquate divers aspects du langage naturel. En particulier, les modèles de langue basés sur les réseaux de neurones ont amélioré l'état de l'art par rapport à la modélisation prédictive de langage, tandis que les *topic models* sont capables de capturer une certaine cohérence thématique. Dans cet article, on explorera comment ces approches peuvent être adaptées et combinées afin de modéliser les aspects linguistiques et littéraires nécessaires pour la génération de poésie. Le système est exclusivement entraîné sur des textes génériques, et sa sortie est contrainte afin de conférer un caractère poétique au vers généré. Le cadre présenté est appliqué à la génération de poèmes en français, et évalué à l'aide d'une évaluation humaine.

ABSTRACT

Automatic Poetry Generation in French

Automatic poetry generation is a challenging task for a computational system. For a poem to be meaningful, both linguistic and literary aspects need to be taken into account. In the last few years, a number of successful approaches have emerged that are able to adequately model various aspects of natural language. Particularly, language models based on neural networks have improved the state of the art with regard to predictive language modeling, while topic models are able to capture some form of thematic coherence. In this article, we will explore how these approaches can be adapted and combined to model the linguistic and literary aspects needed for poetry generation. The system is exclusively trained on generic text, and its output is constrained in order to confer a poetic character to the generated verse. The framework is applied to the generation of poems in French, and it is evaluated using a human evaluation.

MOTS-CLÉS : génération de poésie, réseaux de neurones, factorisation en matrices non-négatives.

KEYWORDS: poetry generation, neural networks, non-negative matrix factorization.

1 Introduction

La génération automatique de poésie est une tâche ardue pour un système informatique. Pour qu'un poème ait du sens, il est important de prendre en compte à la fois des aspects linguistiques et littéraires. Tout d'abord, un système de génération de poésie doit modéliser de manière correcte les phénomènes de langage, tels que la syntaxe, et la cohérence sémantique et discursive. De plus, le système doit intégrer diverses contraintes (telles que la forme et la rime) liées à un genre poétique particulier. Enfin, le système doit faire preuve d'une certaine créativité littéraire, ce qui rend le poème intéressant et digne d'être lu.

Ces dernières années, dans le domaine du traitement automatique des langues, un certain nombre d'approches fructueuses sont apparues, capables de modéliser de manière adéquate divers aspects du langage naturel. En particulier, les modèles de langue basés sur les réseaux de neurones ont amélioré l'état de l'art par rapport à la modélisation prédictive de langage, tandis que les *topic models* sont capables de capturer une certaine forme de cohérence thématique. Dans cet article, on explorera comment ces approches peuvent être adaptées et combinées afin de modéliser les aspects linguistiques et littéraires nécessaires pour la génération de poésie. Plus spécifiquement, dans ce travail, on utilisera des réseaux de neurones récurrents dans une configuration encodeur-décodeur. L'encodeur construit d'abord une représentation d'une phrase entière en incorporant séquentiellement les mots de cette phrase dans un vecteur d'état caché de taille fixe. La représentation finale est ensuite donnée au décodeur, qui émet une séquence de mots selon une distribution de probabilité dérivée de l'état caché de la phrase en entrée. En apprenant au réseau à prédire la phrase suivante avec la phrase actuelle en entrée, le réseau apprend à générer du texte brut avec une certain cohérence discursive. En transformant la distribution de probabilité fournie par le décodeur, afin d'incorporer des contraintes poétiques, le réseau peut être exploité pour la génération de vers poétiques. Il est important de noter que le système de poésie n'est pas entraîné sur des textes poétiques ; au contraire, le système est entraîné sur des textes génériques extraits du web, et ce seront alors les contraintes appliquées qui confèrent un caractère poétique aux vers générés.

Cet article est structuré comme suit. Dans la section 2, on présente un aperçu des travaux connexes sur la génération automatique de poésie. La section 3 décrit ensuite les différentes composantes du système de génération de poésie. Dans la section 4, on présentera un certain nombre d'exemples et une évaluation humaine. La section 5 conclut et examine quelques pistes pour des futurs travaux.

2 Travaux connexes

Il y a une longue et captivante histoire en termes de génération automatique de poésie pour le français (Queneau, 1961; OULIPO, 1981), que l'on qualifierait de créativité mécanique. Au-delà de la simple créativité mécanique, les premières implémentations informatiques se sont souvent appuyées sur des méthodes basées sur des règles ou sur des patrons. L'un des premiers exemples est le système ASPERA (Gervás, 2001) pour l'espagnol, qui repose sur une base de connaissances complexe, un ensemble de règles et un raisonnement à partir de cas. D'autres approches incluent Manurung *et al.* (2012), qui combinent la génération basée sur des règles avec des algorithmes génétiques ; le système de génération PoeTryMe de Gonçalo Oliveira (2012), qui repose sur la génération tabulaire (*chart generation*) et diverses stratégies d'optimisation ; et Veale (2013), qui exploite les expressions métaphoriques en utilisant une approche basée sur les patrons.

Alors que la génération de poésie avec des modèles basés sur des règles et des patrons a une tendance inhérente à être structurellement plutôt rigide, les progrès des méthodes statistiques pour la génération de langage ont ouvert de nouvelles perspectives pour une approche plus variée et hétérogène. Greene *et al.* (2010), par exemple, utilisent un modèle de langage n-gramme en combinaison avec un modèle rythmique implémenté avec des transducteurs à états finis. Et plus récemment, des réseaux de neurones récurrents ont été exploités pour la génération de la poésie. Zhang & Lapata (2014) utilise un encodeur-décodeur RNN pour la génération de poésie chinoise, dans lequel un premier RNN construit une représentation cachée du vers actuel dans un poème, et un deuxième RNN prédit le vers suivant mot par mot, en fonction de la représentation cachée du vers actuel. Le système est entraîné sur un corpus de poèmes chinois. Yan (2016) présente une amélioration de l'approche encodeur-décodeur en incorporant une méthode de raffinement itératif : le réseau construit un poème candidat à chaque itération, et la représentation de l'itération précédente est utilisée lors de la création de la suivante. Et Wang *et al.* (2016) étendent la méthode en utilisant un mécanisme d'attention.

Ghazvininejad *et al.* (2016) combinent des RNNs (afin de modéliser la fluidité syntaxique) avec des calculs de similarité distributionnelle (afin de modéliser la cohérence sémantique) et des automates à états finis (pour imposer des contraintes littéraires telles que le mètre et la rime). Leur système, HAFEZ, est capable de produire des poèmes bien formés avec un raisonnable degré de cohérence sémantique, basés sur un sujet défini par l'utilisateur. Hopkins & Kiela (2017) se concentrent sur les vers rythmiques ; ils combinent un RNN, entraîné sur une représentation phonétique de poèmes, avec une cascade de transducteurs à états finis pondérés. Et Lau *et al.* (2018) présentent un modèle de réseaux de neurones pour la génération de sonnets, qui intègre l'entraînement de la rime et du rythme dans le réseau ; le réseau apprend les motifs de stress iambiques à partir de données, tandis que les paires de mots qui riment sont séparées des paires de mots qui ne riment pas en utilisant une perte basée sur la marge.

Il est à noter que tous les modèles statistiques existants sont entraînés sur un corpus de poésie ; à notre connaissance, notre système est le premier à ne réaliser la génération de poésie qu'avec un modèle exclusivement entraîné sur un corpus générique, ce qui signifie que le caractère poétique est conféré par le modèle lui-même. Deuxièmement, on utilise un modèle sémantique latent pour modéliser la cohérence thématique, ce qui est également nouveau.

3 Modèle

3.1 Architecture neuronale

À la base du système de poésie se trouve un modèle de langage neuronal, entrainé à prédire la phrase suivante S_{i+1} à partir de la phrase courante S_i. L'architecture neuronale est composée de deux réseaux de neurones récurrents à portes (*gated recurrent units*, ou *GRUs* ; Cho *et al.*, 2014) fonctionnant dans une configuration encodeur-décodeur. L'encodeur prend en séquence chaque mot $w^i_{1,\ldots,N}$ de la phrase courante S_i (représenté par son plongement de mot ou *word embedding* $\mathbf{x}$) de manière qu'à chaque pas de temps t_i un état caché $\mathbf{h}_t$ est créé à la base du plongement du mot courant $\mathbf{x}_t$ et l'état caché $\mathbf{h}_{t-1}$ du pas de temps précédent. Pour chaque pas de temps, l'état caché $\hat{\mathbf{h}}_t$ est calculé selon les équations suivantes :

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \hat{\mathbf{h}}_{t-1}) \quad (1)$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \hat{\mathbf{h}}_{t-1}) \quad (2)$$

$$\bar{\mathbf{h}}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \hat{\mathbf{h}}_{t-1})) \quad (3)$$

$$\hat{\mathbf{h}}_t = (1 - \mathbf{z}_t) \odot \hat{\mathbf{h}}_{t-1} + \mathbf{z}_t \odot \bar{\mathbf{h}}_t \quad (4)$$

où $\mathbf{r}_t$ représente la porte de réinitialisation du GRU, $\mathbf{z}_t$ représente la porte de mise à jour, $\bar{\mathbf{h}}_t$ représente le nouveau état candidat, et $\odot$ représente la multiplication élément par élément.

$\mathbf{h}_t$ peut être interprété comme une représentation de la séquence $w_1, \ldots, w_t$, et l'état caché final $\mathbf{h}_N$ sera donc une représentation de la phrase entière. Cet état caché final est ensuite donné comme entrée au décodeur. Le décodeur fait alors une prédiction mot par mot de la phrase suivante, conditionnée sur l'encodeur ; à chaque pas de temps t_{i+1}, le décodeur crée également un état caché $\mathbf{h}_t$ à la base du plongement $\mathbf{x}_t$ du mot courant (prédit par le décodeur dans le pas précédent) et l'état caché $\mathbf{h}_{t-1}$ du pas de temps précédent (le premier état caché étant $\mathbf{h}_N$ qui vient de l'encodeur et le premier mot étant un symbole d'initialisation). Les calculs pour chaque pas de temps $\mathbf{h}_t$ du décodeur sont égaux à ceux utilisés dans l'encodeur (équations 1 à 4).

Afin d'exploiter pleinement la séquence complète de représentations fournie par l'encodeur, l'architecture de base est complétée par un mécanisme d'attention, notamment l'attention dite *générale* (Luong *et al.*, 2015). Le mécanisme d'attention permet au décodeur de consulter l'ensemble des états cachés calculés par l'encodeur ; à chaque pas de temps – pour la génération de chaque mot de la phrase S_{i+1} – le décodeur détermine quels mots de la phrase S_i sont pertinents et sélectionne en conséquence une combinaison linéaire de l'ensemble des états cachés. À cette fin, on calcule d'abord un vecteur d'attention $\mathbf{a}_t$, qui attribue un poids à chaque état masqué $\hat{\mathbf{h}}_i$ produit par l'encodeur (en fonction de l'état caché actuel du décodeur $\mathbf{h}_t$) selon l'équation 5 :

$$\mathbf{a}_t(i) = \frac{\exp(\text{score}(\mathbf{h}_t, \hat{\mathbf{h}}_i))}{\sum_{i'} \exp(\text{score}(\mathbf{h}_t, \hat{\mathbf{h}}_{i'}))} \quad (5)$$

où

$$\text{score}(\mathbf{h}_t, \hat{\mathbf{h}}_i) = \mathbf{h}_t^T \mathbf{W}_\mathbf{a} \hat{\mathbf{h}}_i \quad (6)$$

L'étape suivante consiste à calculer un vecteur de contexte global $\mathbf{c}_t$, qui est une moyenne pondérée (basée sur le vecteur d'attention $\mathbf{a}_t$) de tous les états masqués de l'encodeur. Le vecteur de contexte qui en résulte est ensuite combiné avec l'état caché du décodeur d'origine afin de calculer un nouvel état caché augmenté avec l'attention, $\tilde{\mathbf{h}}_t$:

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_\mathbf{c}[\mathbf{c}_t; \mathbf{h}_t]) \quad (7)$$

où $[\cdot;\cdot]$ représente la concaténation des vecteurs. Enfin, l'état caché qui en résulte $\tilde{\mathbf{h}}_t$ est transformé en distribution de probabilité $p(\mathbf{w}^t | w^{<t}, S_i)$ sur le vocabulaire entier en utilisant une couche softmax.

$$p(\mathbf{w}^t | w^{<t}, S_i) = \text{softmax}(\mathbf{W}_\mathbf{s} \tilde{\mathbf{h}}_t) \quad (8)$$

Comme fonction objective, on optimise la somme des log-probabilités de la phrase suivante, conditionnée sur la représentation cachée de l'encodeur de la phrase actuelle.

$$J_t = \sum_{(S_i, S_{i+1}) \in C} -\log p(S_i | S_{i+1}) \tag{9}$$

Au moment de l'inférence, pour la génération d'un vers, chaque mot est ensuite échantillonné de manière aléatoire en fonction de la distribution de probabilité de sortie. De manière cruciale, le décodeur est entraîné à prédire les mots de la phrase suivante en sens inverse, de sorte que le dernier mot du vers soit le premier généré. Cette opération inverse est importante pour une incorporation efficace de la rime, comme cela sera expliqué dans la section suivante. Une représentation graphique de l'architecture, qui inclut les contraintes discutées ci-dessous, est donnée dans le figure 1.

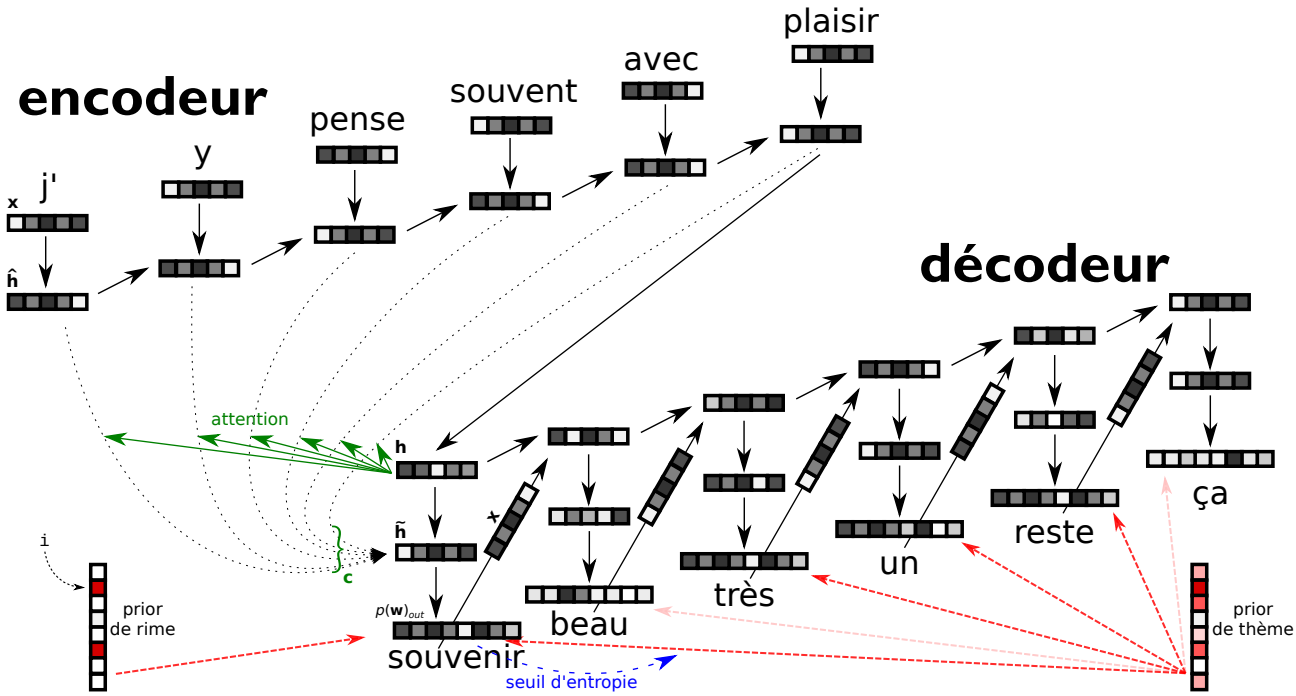

FIGURE 1: Représentation graphique du modèle de génération de poésie. L'encodeur traite le vers actuel mot par mot, et la représentation finale est donnée au décodeur, qui prédit le vers suivant mot par mot, à l'envers. Le mécanisme d'attention est représenté pour le premier pas de temps. La distribution de probabilité *a priori* pour le rime est appliquée au premier pas de temps, et celle pour le thème est facultativement appliquée à tous les pas de temps, en fonction de la valeur d'entropie de la distribution de sortie du réseau.

3.2 Contraintes poétiques comme distributions *a priori*

Étant donné que l'architecture neuronale ci-dessus est entraînée sur des textes génériques, sa sortie ne ressemblera en rien à un poème ; afin de doter la sortie générée d'un certain caractère poétique, on modifiera la distribution de probabilité de sortie du réseau de neurones en appliquant une distribution de probabilité *a priori*. On modélisera deux types de contraintes : une contrainte de rime et une contrainte thématique.

3.2.1 Contrainte de rime

Pour la modélisation de la contrainte de rime, on s'appuie sur une représentation phonétique des mots, extraite de manière automatique depuis le *Wiktionnaire* pour le français. Pour chaque mot, on détermine son rime (c'est-à-dire le groupe de voyelles final, éventuellement suivi d'un groupe de consonnes), ainsi que la groupe de consonnes précédente. Un échantillon de rimes ainsi extraites est donné dans le tableau 1.

mot	rime
reproduit	('dɥ', 'i')
thérapie	('p', 'i')
examen	('m', 'ɛ̃')
canadien	('dj', 'ɛ̃')

TABLE 1: Exemples de rimes extraits du *Wiktionnaire*

L'étape suivante consiste à créer une distribution de probabilité *a priori* pour un son de rime requis :

$$p(\mathbf{w})_{rime} = \frac{1}{Z}\mathbf{x} \text{ avec } \begin{cases} x_i = 1 & \text{if } i \in R \\ x_i = \varepsilon & \text{otherwise} \end{cases} \tag{10}$$

où R est l'ensemble des mots avec le son de rime requis, ε est une valeur très petite pour éviter les erreurs de calcul, et Z est une constante de normalisation pour assurer une distribution de probabilité. On est maintenant en mesure d'appliquer la distribution de probabilité *a priori* afin de repondérer la distribution de probabilité de sortie du réseau de neurones selon la formule 11, chaque fois que le schéma de rimes le requiert :

$$p(\mathbf{w})_{out} = \frac{1}{Z}(p(\mathbf{w}^t|w^{<t}) \odot p(\mathbf{w})_{rime} \tag{11}$$

avec $\odot$ étant la multiplication élément par élément. Rappelons que chaque vers est généré à l'envers ; la repondération par rapport à la rime est appliquée tout au début de la génération, et le mot rime est généré en premier. Cela empêche la génération d'un mot rime maladroit qui ne correspond pas au reste du vers.

3.2.2 Contrainte thématique

Pour la modélisation de la contrainte thématique, on s'appuie sur un modèle de sémantique latente sous forme d'une factorisation en matrices non négatives (NMF ; Lee & Seung, 2001). Des recherches antérieures ont montré que la méthode est capable de produire des dimensions thématiques bien

claires et interprétables (Murphy *et al.*, 2012). Comme entrée, on construit une matrice de fréquence **A**, qui capture les fréquences [1] de co-occurrence des mots du vocabulaire et leurs contextes. Cette matrice est alors factorisée en deux autres matrices non négatives, **W** et **H**.

$$\mathbf{A}_{i\times j} \approx \mathbf{W}_{i\times k}\mathbf{H}_{k\times j} \tag{12}$$

où k est beaucoup plus petit que i, j, de manière que les instances et les traits sont exprimés par un nombre limité de dimensions. De manière cruciale, la factorisation en matrices non négatives impose la contrainte que les trois matrices doivent être non négatives, c'est-à-dire tous les éléments doivent être supérieurs ou égaux à zéro. En utilisant la minimisation de la divergence de Kullback-Leibler comme fonction objective, on veut trouver les matrices **W** et **H** pour lesquelles la divergence entre **A** et **WH** (la multiplication de **W** et **H**) est la plus petite. Cette factorisation est réalisée par l'application itérative de règles de mis à jour. Quelques exemples de dimensions, extraits avec la méthode, sont représentés dans le tableau 2.

dim 1	dim 20	dim 25	dim 90
tendresse	gare	hypocrisie	désespoir
joie	bus	mensonge	terrible
bonheur	métro	accuser	colère
sourires	tram	hypocrite	angoisse
baisers	rer	tort	violente
amour	tgv	arrogance	désarroi
joies	tramway	critiquer	frustration
merveilleux	autoroute	mensonges	souffrance
nostalgie	autobus	bêtises	humiliation
douceur	boulevard	reprocher	impuissance

TABLE 2: Exemples de dimensions thématiques issues de NMF (10 mots les plus saillants)

La factorisation issue du modèle NMF peut être interprétée de manière probabiliste (Gaussier & Goutte, 2005; Ding *et al.*, 2008) : la matrice **W** peut être considérée comme $p(\mathbf{w}|k)$, c'est-à-dire la probabilité d'un certain mot w du vocabulaire, étant donnée la dimension latente k. On pourrait maintenant facilement utiliser cette distribution comme une autre distribution *a priori* thématique, appliquée à chaque sortie ; cependant, une telle modification à l'aveugle de la distribution de probabilité de sortie pour chaque mot de la séquence pose des problèmes par rapport à la structure syntaxique. Pour pallier à cela, on conditionne la modification de la distribution de sortie par le calcul d'une valeur d'entropie sur cette distribution : lorsque l'entropie de la distribution de sortie est faible, le réseau de neurones connaît la choix du mot correct afin de générer une phrase bien formée, donc on ne le changera pas. En revanche, lorsque l'entropie de la distribution de sortie est élevée, on modifie la distribution en utilisant la distribution thématique $p(\mathbf{w}|k)$ d'une dimension latente comme distribution *a priori* (analogue à la formule 11), afin d'insérer la thématique souhaitée. Le seuil d'entropie, au-dessus duquel on utilise la distribution modifiée, est défini expérimentalement.

Notez que la contrainte de rime et la contrainte thématique peuvent facilement être combinées afin de générer un mot de rime thématique, en multipliant les trois distributions concernées, puis en procédant à une normalisation.

1. Les fréquences brutes sont pondérées en utilisant l'information mutuelle spécifique (*pointwise mutual information* ; Bullinaria & Levy, 2007 ; Turney & Pantel, 2010).

3.3 Cadre d'optimisation global

La génération d'un vers est réalisé dans un cadre d'optimisation global. On intègre le modèle de génération dans un cadre d'optimisation pour deux raisons. Premièrement, la génération d'un vers est un processus d'échantillonnage, sujet au hasard. Le cadre d'optimisation nous permet de choisir le meilleur échantillon en fonction des contraintes définies ci-dessus. Deuxièmement, l'optimisation nous permet de définir quelques critères supplémentaires qui aident dans la sélection du meilleur vers. Pour chaque vers final généré, le modèle génère un nombre considérable de vers candidats ; chaque candidat est alors noté en fonction des critères suivants :

— la log-probabilité du vers généré, en fonction de l'architecture encodeur-décodeur (section 3.1) ;
— respect de la contrainte de rime (section 3.2.1) ; de plus, l'extraction du groupe de consonnes précédent (cf. tableau 1) permet de donner un score plus élevé aux mots rimes avec des groupes de consonnes précédents disparates, ce qui permet d'obtenir des rimes plus intéressantes ;
— respect de la contrainte thématique (section 3.2.2) ; le score est modélisé comme la somme des probabilités de tous les mots pour la dimension définie ;
— le nombre optimal de syllabes, modélisé comme une distribution gaussienne avec une moyenne μ et un écart-type σ ; [2]
— la log-probabilité d'un modèle de n-grammes standard.

Le score de chaque critère est normalisé à l'intervalle $[0, 1]$ à l'aide d'un normalisation *min-max*, et la moyenne harmonique [3] de tous les scores est considérée comme le score final de chaque candidat. Après la génération d'un nombre prédéfini de candidats, le candidat avec le score optimal est conservé et ajouté au poème.

4 Résultats et évaluation

4.1 Détails de mise en œuvre

L'architecture neuronale a été entraînée sur un corpus de textes web en français à caractère général, construit à base du corpus CommonCrawl [4]. Le corpus dans son intégralité contient 11 milliards de mots ; cependant, on effectue un certain nombre d'étapes de filtrage afin de ne conserver que des paires de phrases propres :

— on ne garde que des phrases de 20 mots maximum ;
— on ne garde que des phrases qui contiennent au moins un mot fonction (par exemple, les pronoms communs) d'une liste prédéfinie, l'idée étant de ne garder que des vraies phrases et de filtrer le bruit ;
— de toutes les phrases qui restent après les deux premières étapes de filtrage, on ne garde que les phrases qui apparaissent successivement dans un document.

2. On a également mené des expériences avec des contraintes basées sur le mètre et les pieds de vers, mais les premières expériences indiquaient que le système avait tendance à produire des vers très rigides. Un simple comptage des syllabes tend à donner une variation plus intéressante.

3. La moyenne harmonique est calculée par $\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$; elle est choisie pour balancer les différents scores.

4. `commoncrawl.org`

Après filtrage, la taille du corpus est réduite à 400 million de mots. On utilise un vocabulaire de 15 000 mots (sélectionnés par rapport à leur fréquence) ; au-delà, les mots sont remplacés par un token <unk> (dont la probabilité est fixée à zero pendant la phase de génération).

L'encodeur ainsi que le décodeur sont tous les deux constitués de deux couches de GRUs avec un état caché de 2048, et la taille de plongements de mots est de 512 ; les plongements d'encodeur, de décodeur, et de sortie sont partagés (Press & Wolf, 2017). On optimise les paramètres du modèle en utilisant une descente de gradient stochastique, partant d'un taux d'apprentissage de 0,2, qui est divisé par 4 lorsque la fonction de coût n'améliore plus sur un ensemble de validation. On utilise un *batch size* de 64, et on applique du *gradient clipping*. L'architecture neuronale a été implémentée en utilisant PyTorch (Paszke *et al.*, 2017), en s'appuyant considérablement sur le module OpenNMT (Klein *et al.*, 2017). Par rapport à la contrainte thématique, on utilise un seuil d'entropie de 2,70.

Le modèle n-gramme utilisé est un modèle standard d'ordre 3 lissé par Kneser-Ney, entrainé en utilisant *KenLM* (Heafield, 2011). Le modèle NMF est factorisé en 100 dimensions, la matrice de fréquences étant construite avec une fenêtre de phrases, et en utilisant la divergence de Kullback-Leibler comme objective. Le modèle n-gramme ainsi que le modèle NMF sont entrainés sur l'intégralité du corpus sans filtrage. Pour la contrainte de nombre de syllabes, on utilise $\mu = 12, \sigma = 2$.

On génère environ 2000 candidats pour chaque vers, selon un schéma de rimes fixe (ABAB CDCD). Quatre exemples représentatifs de poèmes générés par le système sont montré dans la figure 2. Notez qu'aucune sélection humaine n'a été effectuée sur les poèmes utilisés pour l'évaluation ; tous les poèmes ont été générés en une seule fois, sans *cherry picking*.

4.2 Évaluation humaine

L'évaluation quantitative de la créativité est loin d'être simple, et cela n'est pas moins vrai pour les artefacts créatifs qui sont générés de manière automatique. Des mesures d'évaluation automatique qui calculent la similarité de la sortie du système avec des textes de référence standard (telles que BLEU ou ROUGE), et qui pourraient être utilisés pour évaluer les tâches de génération standard, peuvent difficilement être qualifiées d'appropriées quand il s'agit de génération créative. C'est l'une des raisons pour lesquelles la plupart des chercheurs ont fait recours à une évaluation humaine, bien qu'il faille garder à l'esprit que l'évaluation de créativité textuelle est par nature subjective, en particulier en ce qui concerne la valeur poétique. Pour une discussion sur le sujet et un aperçu des différentes méthodes d'évaluation, voir Gonçalo Oliveira (2017).

Dans cette recherche, on adopte le cadre d'évaluation de Zhang & Lapata (2014), dans lequel il est demandé aux annotateurs d'évaluer les poèmes sur une échelle de cinq points, en fonction d'un certain nombre de caractéristiques, à savoir :

— *fluidité* : le poème est-il grammatical et syntaxiquement bien formé ?
— *cohérence* : le poème est-il structuré thématiquement ?
— *signification* : le poème transmet-il un message significatif au lecteur ? Le poème a-t-il un sens ?
— *caractère poétique* : le texte affiche-t-il les caractéristiques d'un poème ?

En plus, on demande aux annotateurs de juger si le poème est écrit par un humain ou un ordinateur.

Au total, on évalue six ensembles de poèmes différents, issus de différentes instanciations de modèles. Les différents ensembles de poèmes pris en compte lors de l'évaluation sont les suivants :

Malgré mon enthousiasme, le chagrin s'allonge
Le bonheur est toujours superbe
Toi, tu es un merveilleux songe
Je te vois rêver de bonheur dans l'herbe

Tu trouveras le bonheur de tes rêves
Je t'aime comme tout le monde
Je t'aime mon amour, je me lève
Je ressens pour toi une joie profonde

~

La route vers la ville est imprenable
Nous décidons de prendre le bus vers le tram
De plus la station de métro est très agréable
Je suis en train de rentrer dans ma rame

La gare, plusieurs personnes m'observent
Je suis allée dans la rue des portes
Je m'aperçois que le tgv, ça énerve
Si je la voie, c'est que la sncf est morte

~

Rien ne prouve qu'il s'indigne
Dans le cas contraire, ce n'est pas grave
Si la vérité est fausse, c'est très mauvais signe
Il est vrai que les gens le savent

Et cela est faux, mais qu'importe
En fait, le mensonge, c'est l'effroi
La négation de l'homme en quelque sorte
Le tort n'est pas de penser cela, il est magistrat

~

Hélas, après sa mort, ce fut elle qui cède
Ce fut un moment d'une angoisse extrême
Un sentiment d'incompréhension, mais sans remède
Une peur en colère, et parfois même

Un véritable sentiment de panique
Ce qui provoque une rage étrange
Il s'ensuit un drame tragique
On sent la tragédie qui, sans excès, s'arrange

FIGURE 2: Quatre exemples représentatifs de poèmes générés par le système ; les poèmes, de haut en bas, ont été générés respectivement en utilisant la dimension 1, 20, 25, et 90 (cf. tableau 2).

1. *random* : des poèmes générés par un modèle de référence aléatoire où, pour chaque vers, on sélectionne de manière aléatoire une phrase, entre 7 et 15 mots, dans un grand corpus ; l'idée est que les phrases sélectionnées par le modèle de référence seront assez fluides (puisqu'elles proviennent d'un corpus réel), mais sans cohérence (en raison de leur sélection aléatoire) ;
2. *rnn* : des poèmes générés par l'architecture neuronale décrit en section 3.1, sans aucune contrainte supplémentaire ;
3. *rime* : des poèmes générés par l'architecture neuronale, augmenté avec la contrainte de rime ;
4. nmf_{rand} : des poèmes générés par l'architecture neuronale, augmentée à la fois avec la contrainte de rime et la contrainte thématique, où l'une des dimensions NMF (induites de manière automatique) est sélectionnée de manière aléatoire ;
5. nmf_{spec} : des poèmes générés par l'architecture neuronale, augmentée à la fois avec la contrainte de rime et la contrainte thématique, où l'une des dimensions NMF (induites de manière automatique) est spécifiée manuellement [5] ;
6. *humain* : poèmes écrits par des humains [6].

22 annotateurs ont évalué 30 poèmes au total (5 pour chacun des six modèles évalués), de sorte que chaque poème a été évalué par au moins 4 annotateurs. Les résultats sont présentés dans le tableau 3.

modèle	fluidité	cohérence	signification	caractère poétique	écrit par humain (%)
random	2,95	1,86	1,68	2,18	0,00
rnn	3,45	2,73	2,59	2,55	0,27
rime	**3,82**	2,55	2,18	3,23	0,14
nmf_{rand}	3,64	3,32	3,09	2,86	0,27
nmf_{spec}	**3,82**	**3,82**	**3,55**	**3,95**	**0,45**
humain	4,59	4,59	4,50	4,81	0,95

TABLE 3: Résultats de l'évaluation humaine (score moyenne pour tous les annotateurs)

Tout d'abord, on remarque que tous les modèles fonctionnent mieux que le modèle de référence aléatoire, même en ce qui concerne la fluidité syntaxique ($p < 0,01$ en utilisant un test de permutation bilatéral ; notez que le modèle de référence est constituée de phrases réelles). Les bons scores obtenus pour nos modèles avec contraintes (*rime* et nmf_*) indiquent que l'application de contraintes ne nuit pas à la grammaticalité des vers. Deuxièmement, on constate que la contrainte de rime améliore le caractère poétique ($p < 0,05$ vis-à-vis *rnn*), et que la contrainte thématique améliore à la fois la cohérence ($p < 0,05$) et la signification ($p < 0,01$). On note également que le score pour caractère poétique est considérablement plus élevé ($p < 0,01$) pour nmf_{spec} (avec un thème qu'on pourrait juger poétique) que pour nmf_{rand} (avec un thème aléatoire, que l'on considérerait souvent comme plus banal). Finalement, on constate que les meilleurs scores par rapport à tous les critères sont obtenus avec le modèle nmf_{spec}, pour lesquels les poèmes sont jugés être écrits par un humain dans presque la moitié des cas.

5. Ceci peut être considéré comme définissant manuellement le thème du poème généré. La dimension spécifiée est sélectionnée pour son caractère poétique. On prend la dimension 1 du tableau 2 comme dimension spécifiée.

6. On prend des poèmes avec le même schéma de rimes que les poèmes générés, parmi les poèmes les mieux classés sur le site `short-edition.com`.

5 Conclusion

Dans cet article, on a présenté un système pour la génération de poésie en français. On utilise des réseaux de neurones en configuration encodeur-décodeur pour générer des vers candidats, en modifiant la distribution de sortie pour incorporer des contraintes littéraires et thématiques. Dans un cadre d'optimisation, on sélectionne alors parmi un nombre de candidats le meilleur vers pour inclusion dans le poème. Les résultats d'une évaluation humaine indiquent que le système est capable de générer des poèmes crédibles, avec des bons scores en termes de fluidité et de cohérence, ainsi qu'en termes de signification et de caractère poétique. Dans notre meilleure configuration, presque la moitié des poèmes générés sont jugés être écrits par un humain. La méthode présentée est générale, ce qui signifie qu'elle peut facilement être étendue à d'autres langues.

Afin de permettre l'utilisation et l'expérimentation ainsi que l'inspection du modèle, le système de génération de poésie est mis à disposition sous forme de logiciel open source. La version actuelle est téléchargéable en utilisant le lien `https://github.com/timvdc/poetry`.

On conclue avec quelques pistes pour des travaux futurs. Tout d'abord, on aimerait explorer différentes architectures de réseaux de neurones. Plus précisément, on pense que des approches hiérarchiques (Serban *et al.*, 2017) ainsi que le réseau dite *transformer* (Vaswani *et al.*, 2017) conviendraient particulièrement à la génération de la poésie. Deuxièmement, on aimerait incorporer d'autres dispositifs poétiques, notamment ceux basés sur le sens. La poésie captivante repose souvent sur l'utilisation d'un langage figuré, tel que le symbolisme et la métaphore. Une incorporation spécifique de tels dispositifs signifierait un pas important vers une génération de poésie vraiment inspirée.

Références

BULLINARIA J. A. & LEVY J. P. (2007). Extracting semantic representations from word co-occurrence statistics : A computational study. *Behavior research methods*, **39**(3), 510–526.

CHO K., VAN MERRIENBOER 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 : Association for Computational Linguistics.

DING C., LI T. & PENG W. (2008). On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, **52**(8), 3913–3927.

GAUSSIER E. & GOUTTE C. (2005). Relation between plsa and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 601–602 : ACM.

GERVÁS P. (2001). An expert system for the composition of formal spanish poetry. In *Applications and Innovations in Intelligent Systems VIII*, p. 19–32, London : Springer.

GHAZVININEJAD M., SHI X., CHOI Y. & KNIGHT K. (2016). Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1183–1191, Austin, Texas : Association for Computational Linguistics.

GONÇALO OLIVEIRA H. (2012). Poetryme : a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, **1**, 21.

GONÇALO OLIVEIRA H. (2017). A survey on intelligent poetry generation : Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, p. 11–20.

GREENE E., BODRUMLU T. & KNIGHT K. (2010). Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, p. 524–533 : Association for Computational Linguistics.

HEAFIELD K. (2011). KenLM : faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, p. 187–197, Edinburgh, Scotland, United Kingdom.

HOPKINS J. & KIELA D. (2017). Automatically generating rhythmic verse with neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 168–178 : Association for Computational Linguistics.

KLEIN G., KIM Y., DENG Y., SENELLART J. & RUSH A. (2017). Opennmt : Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, p. 67–72 : Association for Computational Linguistics.

LAU J. H., COHN T., BALDWIN T., BROOKE J. & HAMMOND A. (2018). Deep-speare : A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1948–1958 : Association for Computational Linguistics.

LEE D. D. & SEUNG H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, p. 556–562.

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 : Association for Computational Linguistics.

MANURUNG R., RITCHIE G. & THOMPSON H. (2012). Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence*, **24**(1), 43–64.

MURPHY B., TALUKDAR P. & MITCHELL T. (2012). Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012*, p. 1933–1950 : The COLING 2012 Organizing Committee.

OULIPO (1981). *Atlas de Littérature Potentielle*. Paris, France : Gallimard.

PASZKE A., GROSS S., CHINTALA S., CHANAN G., YANG E., DEVITO Z., LIN Z., DESMAISON A., ANTIGA L. & LERER A. (2017). Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems*.

PRESS O. & WOLF L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, p. 157–163 : Association for Computational Linguistics.

QUENEAU R. (1961). *Cent mille milliards de poèmes*. Paris, France : Gallimard.

SERBAN I. V., SORDONI A., LOWE R., CHARLIN L., PINEAU J., COURVILLE A. & BENGIO Y. (2017). A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

TURNEY P. D. & PANTEL P. (2010). From frequency to meaning : Vector space models of semantics. *Journal of artificial intelligence research*, **37**, 141–188.

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.

VEALE T. (2013). Less rhyme, more reason : Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the international conference on computational creativity*, p. 152–159.

WANG Q., LUO T., WANG D. & XING C. (2016). Chinese song iambics generation with neural attention-based model. In *Proceedings of International Joint Conference on Artificial Intelligence*, p. 2943–2949.

YAN R. (2016). i, poet : Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of International Joint Conference on Artificial Intelligence*, p. 2238–2244.

ZHANG X. & LAPATA M. (2014). Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 670–680 : Association for Computational Linguistics.

Modèles neuronaux hybrides pour la modélisation de séquences : le meilleur de trois mondes

Marco Dinarelli[1] Loïc Grobol[2,3]
(1) LIG, Bâtiment IMAG - 700 avenue Centrale - Domaine Universitaire de Saint-Martin-d'H'eres
(2) Lattice CNRS, 1 rue Maurice Arnoux, 92120 Montrouge, France
(3) ALMAnaCH Inria, 2 rue Simone Iff, 75589 Paris, France
marco.dinarelli@univ-grenoble-alpes.fr, loic.grobol@gmail.com

RÉSUMÉ
Nous proposons une architecture neuronale avec les caractéristiques principales des modèles neuronaux de ces dernières années : les réseaux neuronaux récurrents bidirectionnels, les modèles *encodeur-décodeur*, et le modèle *Transformer*. Nous évaluons nos modèles sur trois tâches d'étiquetage de séquence, avec des résultats aux environs de l'état de l'art et souvent meilleurs, montrant ainsi l'intérêt de cette architecture hybride pour ce type de tâches.

ABSTRACT
Hybrid Neural Networks for Sequence Modelling : The Best of Three Worlds

We propose a neural architecture with the main characteristics of the most successful neural models of the last years : bidirectional RNNs, *encoder-decoder*, and the *Transformer* model. Evaluation on three sequence labelling tasks yields results that are close to the state-of-the-art for all tasks and better than it for some of them, showing the pertinence of this hybrid architecture for this kind of tasks.

MOTS-CLÉS : Réseaux neuronaux, modélisation de séquences, MEDIA, WSJ, TIGER.

KEYWORDS : Neural Networks, sequence modelling, MEDIA, WSJ, TIGER.

1 Introduction

L'étiquetage de séquences est un problème important du TAL, de nombreux problèmes pouvant être modélisés comme des étiquetage de séquences. Les cas plus classiques sont l'étiquetage en parties du discours (*POS tagging*), la segmentation syntaxique, la reconnaissance d'entités nommées (Collobert *et al.*, 2011), ou encore la compréhension automatique de la parole dans les systèmes de dialogue humain-machine (De Mori *et al.*, 2008).

D'autres problèmes de TAL peuvent être divisés en plusieurs étapes, dont la première peut être modélisée comme étiquetage de séquences. Nous plaçons dans cette catégorie de problèmes l'analyse syntaxique, qui peut être décomposée en étiquetage en parties du discours et en analyse des constituants (Collins, 1997) ; la détection de chaînes de coréférences (Soon *et al.*, 2001 ; Ng & Cardie, 2002), qui peut être décomposée en détection de mentions et détection des mentions coréférentes ; mais aussi la détection d'entités nommées étendues (Grouin *et al.*, 2011 ; Dinarelli & Rosset, 2012a,b)

Plus largement, la traduction automatique et l'analyse syntaxique peuvent également être traitées comme des problèmes de prédiction de séquences bout-en-bout (Sutskever *et al.*, 2014 ; Bahdanau

et al., 2014 ; Vaswani *et al.*, 2017 ; Vinyals *et al.*, 2015), ainsi qu'une large classe de tâches de compréhension du langage (Devlin *et al.*, 2018). Il serait donc possible d'utiliser un modèle pour la prédiction de séquences dans un cadre d'apprentissage multi-tâche pour traiter la plupart des tâches de TAL, ce qui montre l'intérêt de la recherche d'architectures alternatives pour ce type de modèles

Les tâches en plusieurs étapes peuvent également être traitées de bout-en-bout par un modèle unique — comme l'analyse syntaxique en constituants, qui est typiquement traitée par un modèle qui effectue à la fois l'étiquetage en partie du discours et l'analyse syntaxique (Rush *et al.*, 2012). Cependant, même dans ce cas un pré-apprentissage de représentations par des réseaux neuronaux récurrents et leur ajustement sur des tâches plus simple — dont des tâches d'étiquetage de séquences — peut améliorer considérablement les performances (Peters *et al.*, 2018). On peut également rapprocher ce procédé du pré-apprentissage de prolongements lexicaux, dont l'efficacité n'est plus à prouver (Lample *et al.*, 2016 ; Ma & Hovy, 2016).

Dans cet article nous nous limitons à proposer une architecture neuronale pour la modélisation de séquences dans un sens plus classique, c'est à dire l'étiquetage en partie du discours, l'analyse morpho-syntaxique, et l'étiquetage en concepts sémantiques tel qu'il est réalisé pour la tâche de compréhension de la parole dans le cadre du dialogue humain-machine (De Mori *et al.*, 2008).

En nous inspirant de (Chen *et al.*, 2018), duquel nous nous sommes inspirés également pour le titre de notre article, qui propose des modèles hybrides entre *Encodeur-décodeur* et *Transformer*, nous proposons une architecture avec les caractéristiques principales des modèles neuronaux plus efficaces proposés ces dernières années : les modèles RNNs bidirectionnels (Lample *et al.*, 2016 ; Ma & Hovy, 2016), les modèles *Encodeur-décodeur* (Sutskever *et al.*, 2014 ; Bahdanau *et al.*, 2014) et le modèle *Transformer* (Vaswani *et al.*, 2017).

Nous évaluons notre architecture sur trois tâches classiques d'étiquetage de séquences : la compréhension de la parole en français (corpus MEDIA) (Bonneau-Maynard *et al.*, 2006), le *POS tagging* en anglais (corpus WSJ) (Marcus *et al.*, 1993), et l'analyse morpho-syntaxique en allemand (corpus TIGER) (Brants *et al.*, 2004). Les résultats dépassent souvent l'état-de-l'art, et ils nous permettent de conclure dans tous les cas que notre architecture a sa place parmi les modèles neuronaux. L'adaptation de nos modèles à la traduction automatique et à l'analyse syntaxique est laissée comme futur travail.

2 Architectures Neuronales

Les modèles neuronaux proposés s'inspirent des modèles *LSTM+CRF* (Lample *et al.*, 2016 ; Ma & Hovy, 2016) pour encoder l'information en entrée ; des modèles *Encodeur-décodeur* pour l'architecture globale, et des modèles proposés dans (Dinarelli & Tellier, 2016a,b ; Dinarelli *et al.*, 2017 ; Dupont *et al.*, 2017) pour une prise de décision bidirectionnelle au niveau des unités de sortie (les étiquettes). À cette architecture nous avons ensuite ajouté certains caractéristiques du modèle *Transformer*.

2.1 Encodeur

L'encodeur de notre réseau est une couche cachée bidirectionnelle de type GRU (Cho *et al.*, 2014) qui prend en entrée une séquence S^{lex} représentant les mots par la concaténation de plongements de

mots non-contextuels ($E_w(w_i)$) et de représentations issues de leurs caractères $h_c(w_i)$. Celle-ci est calculée par un réseau de neurones récurrent dédié similaire à celui utilisé par Ma & Hovy (2016), la seule différence étant que nous utilisons une couche GRU, au lieu d'un LSTM.

La représentation des mots au niveau des caractères pour un mot quelconque w est calculée comme dans (Dinarelli & Grobol, 2019) :

$$\begin{aligned} S^c(w) &= (E_c(c_{w,1}), \dots, E_c(c_{w,n})) \\ (h_c(c_{w,1}), \dots, h_c(c_{w,n})) &= \text{GRU}_c(S^c(w), h_c^0) \\ h_c(w) &= \text{FFNN}(Sum(h_c(c_{w,1}), \dots, h_c(c_{w,n}))) \end{aligned} \tag{1}$$

Avec les notations suivantes : E_c pour les plongements des caractères, $c_{w,i}$ pour le i-eme caractère du mot w, $S^c(w)$ pour la séquence de plongements des caractères du mot w, GRU_c pour la couche GRU pour les caractères, $h_c(c_{w,i})$ pour l'état caché associé au i-eme caractère du mot w. GRU_c, comme GRU_w, est une couche GRU bidirectionnelle.

La représentation cachée d'un mot w_i est ensuite calculée comme suit :

$$\begin{aligned} S^{lex} &= ([E_w(w_1), h_c(w_1)], \dots, [E_w(w_N), h_c(w_N)]) \\ (h_{w_1}, \dots, h_{w_N}) &= \text{GRU}_w(S^{lex}, h_w^0) \end{aligned} \tag{2}$$

Puisque la couche GRU_w parcourt la séquence en avant et en arrière, h_{w_i} dépend ainsi à la fois du mot w_i et de son contexte.Quand des traits additionnels sont disponibles en entrée, ils sont plongés de la même façon que les mots et concaténés à ces derniers dans la séquence S^{lex}.

2.2 Décodeurs

Notre modèle utilise une représentation des contextes d'étiquettes gauches et droits comme proposé par Dinarelli *et al.* (2017) ; Dupont *et al.* (2017). À la place des couches cachées linéaire nous utilisons cependant des couches récurrentes de type GRU. Nous utilisons une couche $\overleftarrow{\text{GRU}}_e$ *backward* pour encoder le contexte droit, et une couche $\overrightarrow{\text{GRU}}_e$ *forward* pour le contexte gauche. Ces couches prennent en entrée à la fois la représentation de l'information lexicale calculée par l'encodeur et les plongements des étiquettes $E_e(e_i)$, ce qui les rend similaires au décodeur utilisé dans l'architecture originale proposée par Sutskever *et al.* (2014) ; Bahdanau *et al.* (2014). Une évolution par rapport à cette architecture est notre utilisation de deux décodeurs, un pour le contexte droit et un pour le contexte gauche.

Le calcul du contexte droit par le décodeur *backward* $\overleftarrow{\text{GRU}}_e$ se fait formellement comme suit :

$$\overleftarrow{h_{e_i}} = \overleftarrow{\text{GRU}}_e([h_{w_i}, E_e(e_{i+1})], \overleftarrow{h_{e_{i+1}}}) \tag{3}$$

Le calcul du contexte gauche $\overrightarrow{h_{e_i}}$ est fait de façon similaire par le décodeur *forward* $\overrightarrow{\text{GRU}}_e$.

2.3 Couche de sortie

Afin que le modèle puisse prendre une décision globale, nous ajoutons une couche de sortie sur le décodeur *backward* composé d'une couche cachée linéaire suivie d'une fonction *log-softmax* qui

calcule les log-probabilités des prédictions *backward* :

$$\begin{aligned} \overleftarrow{\text{log-P}}(e_i) &= \text{log-softmax}(W_{bw}[h_{w_i}, \overleftarrow{h_{e_i}}] + b_{bw}) \\ e_i &= \text{argmax}(\overleftarrow{\text{log-P}}(e_i)) \end{aligned} \tag{4}$$

Le décodeur *forward* prédit les étiquettes en utilisant à la fois les contextes d'étiquettes $\overrightarrow{h_{e_i}}$ et $\overleftarrow{h_{e_i}}$, ainsi que l'information lexicale h_{w_i} calculée par l'encodeur :

$$\begin{aligned} \overrightarrow{\text{log-P}}(e_i) &= \text{log-softmax}(W_o[\overrightarrow{h_{e_i}}, h_{w_i}, \overleftarrow{h_{e_i}}] + b_o) \\ e_i &= \text{argmax}(\overrightarrow{\text{log-P}}(e_i)) \end{aligned} \tag{5}$$

Pour renforcer le caractère global de la décision, la log-probabilité de la sortie finale est calculée comme la moyenne arithmétique des deux sorties *forward* et *backward* : $\frac{1}{2}(\overrightarrow{\text{log-P}}(e_i) + \overleftarrow{\text{log-P}}(e_i))$. [1]. Ceci permet d'inciter le modèle à fournir des prédictions de qualité dès la phase *backward* plutôt que de s'y contenter d'heuristiques grossières, voire de se reposer uniquement sur les prédictions de la phase *forward*.

2.4 Apprentissage

Tous nos modèles sont appris en minimisant l'opposé de la *log-vraisemblance* $\mathcal{LL}$ sur les données d'apprentissage. Formellement :

$$-\mathcal{LL}(\Theta|D) = -\sum_{d=1}^{|D|}\sum_{i=1}^{N_d}\frac{1}{2}(\text{log-p}(\overrightarrow{e_i}) + \text{log-p}(\overleftarrow{e_i})) + \frac{\lambda}{2}|\Theta|^2) \tag{6}$$

La première somme parcourt les données D de taille $|D|$, alors que la seconde somme parcourt chaque exemple d'apprentissage S_d, de longueur N_d.

Puisque les données utilisées dans ce travail ont une taille relativement petite, et nos modèles sont relativement complexes, nous ajoutons en terme de régularisation L_2 à la fonction de coût, dont λ constitue le coefficient.

2.5 Le meilleur de trois mondes

Le modèle décrit jusqu'ici reprend le principe introduit par (Dinarelli *et al.*, 2017 ; Dupont *et al.*, 2017) de prédire les étiquettes à partir d'une représentation des mots tenant compte des contextes gauche et droit aussi bien pour les mots que pour les étiquettes Notre modèle réunit également les caractéristiques d'un RNN bidirectionnel, et du modèle *encodeur-décodeur*. Nous utilisons de plus deux décodeurs au lieu d'un seul comme dans l'architecture originale.

En partant de ce modèle nous avons ajouté certaines des caractéristiques du modèle *Transformer*.

Nous notons que l'article duquel nous nous sommes inspiré pour ce travail (Chen *et al.*, 2018), analysait la combinaison de deux architecture de type *encodeur-décodeur* , l'une récurrente (Sutskever

1. Ce qui équivaut au logarithme de la moyenne géométrique des probabilités

et al., 2014 ; Bahdanau *et al.*, 2014), l'autre basée sur le modèle *Transformer*, donc non-récurrente. Ce travail nous montre qu'une telle combinaison est possible mais aussi avantageuses par rapport aux deux architecture de départ.

Dans ce travail nous avons décidé de combiner trois architectures non pas pour des raisons de complémentarité, mais pour utiliser les forces de chacune d'entre elles, et pour pallier leur faiblesses. Dans les tâches sur lesquelles nous nous évaluons dans ce travail, il y a une correspondance un-à-un entre les unités d'entrée (les mots) et les unités de sortie (les étiquettes). Nos expériences préliminaires de détection de mentions pour la coréférence avec un *Transformer*, ainsi que certains résultats de la littérature (Guo *et al.*, 2019), suggère que se passer de cette correspondance conduit à des pertes remarquables de performance. [2] L'encodeur et les décodeurs de notre architecture utilisent pour cette raison l'information de correspondance un-à-un des tâches d'étiquetages de séquences. Nous sommes conscient cependant que cette information n'est pas utilisable dans des tâches comme la traduction automatique ou l'analyse syntaxique, auxquelles nous adapterons nos modèles dans le future.

Nous avons choisi d'utiliser une architecture *encodeur-décodeur* puisque nous avons montré que la prise en compte d'un contexte au niveau des étiquettes dans un décodeur est plus efficace qu'utiliser une couche CRF neuronale (Dinarelli *et al.*, 2017), ceci sera montré également dans ce travail. De plus, nous utilisons deux décodeurs, ce qui améliore encore les capacités de modélisation de contextes au niveau des unités de sortie. Cette solution n'est pas possible avec un *Transformer* classique.

Enfin, nous avons décidé d'intégrer certaines caractéristiques du *Transformer* pour pallier les limitations des RNNs concernant leur difficulté dans l'apprentissage, liée au problème de la longueur des parcours du signal d'apprentissage dans la phase de propagation en arrière (voir plus bas, mais aussi plus en détail (Vaswani *et al.*, 2017)).

Le modèle *Transformer* originel (Vaswani *et al.*, 2017) présentait une alternative aux réseaux récurrents fondée sur un mécanisme d'attention à têtes multiples (*Multi-Head Attention*). Des travaux récents (Dehghani *et al.*, 2018 ; Dai *et al.*, 2019) suggèrent cependant que ses performances peuvent être améliorées par l'ajout de certaines formes de récurrence.

Une autre caractéristique intéressante du *Transformer* est l'utilisation des connexions résiduelles (*Skip connections*). Celles-ci permettent de mitiger le problème classique d'évanouissement du gradient dans les réseaux neuronaux profonds. Dans le cas des RNN, l'utilisation de mécanismes de portes permet déjà de limiter ce phénomène, mais nos expériences suggèrent que pour des séquences très longues, elle ne suffit pas à s'en affranchir complètement.

Le mécanisme d'attention ne présente pas ce problème. Chaque élément de la séquence en entrée étant reliée à un nombre fixe de couches, le parcours de rétro-propagation du signal d'apprentissage est assez court par rapport aux RNNs. De plus, l'utilisation des connexions résiduelles renforce davantage la puissance du signal rétro-propagé, permettant à celui-ci de sauter une couche, et donc son affaiblissement, à chaque fois qu'une connexion résiduelle est utilisée.

Nous pouvons interpréter chaque bloc d'une architecture *Transformer*, que ce soit l'encodeur ou le décodeur, comme étant composé par un sous-module qui *"encode"* des traits contextuels, que nous appelons avec abus de langage *Encoder*, et par un sous-module *feed-forward* qui *"transforme"* ces traits en les plongeant dans un espace de traits "profonds". Dans l'architecture *Transformer* (Vaswani *et al.*, 2017) la sortie des deux sous-modules est additionnée à l'entrée du sous-module (*skip connection*) et normalisée au niveau de la couche (*layer normalisation*). Cette interprétation est montrée

2. Nous obtenons plus que 3 points de F-mesure en moins avec le *Transformer*

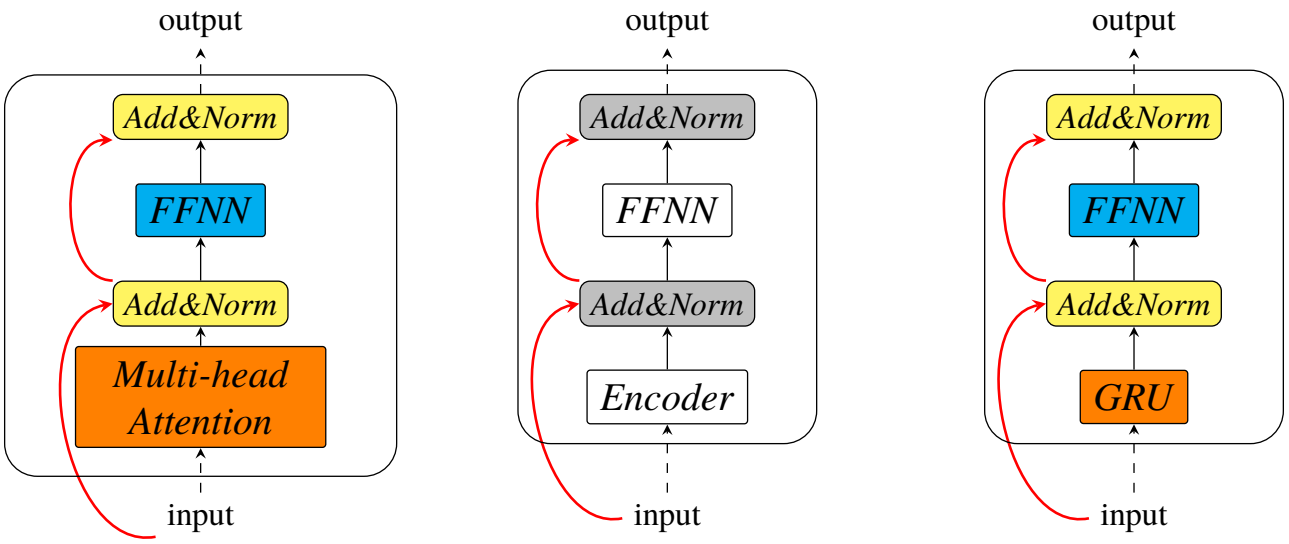

FIGURE 1 – Architecture générale, d'un point de vue conceptuel, de chaque bloc d'un réseau *Transformer* (au centre), sa réalisation pratique dans un modèle *Transformer* (à gauche) et sa réalisation dans notre réseau (à droite), dans lequel nous utilisons une couche GRU pour encoder des traits contextuels

dans la figure 1 au centre. Les connexions résiduelles sont montrées en rouge.

Le *Transformer* présenté par Vaswani *et al.* (2017) instancie l'architecture générique décrite ci-dessus en utilisant comme *Encoder* le mécanisme d'attention à têtes multiple. Cette architecture est présentée à gauche dans la figure 1.

Nous avons modifié notre architecture neuronale pour qu'elle instancie également l'architecture générique décrite plus haut. Dans notre architecture nous utilisons une couche récurrente GRU comme *Encoder*, et exactement le même réseau *Feed-Forward* que dans l'architecture proposée par Vaswani *et al.* (2017), c'est à dire avec deux couches. Notre architecture est montrée à droite dans la figure 1.

Les caractéristiques du modèle *Transformer* que nous intégrons dans notre modèle sont donc les *skip-connections* et la normalisation au niveau des couches cachées (*Add&Norm* dans la figure 1), ainsi que le réseau *Feed-Forward* qui re-encode la sortie des couches GRU (*FFNN* dans la figure 1). Ainsi, en suivant la même chaîne d'opérations suggérée par Chen *et al.* (2018), la sortie de l'encodeur h_{w_i} est calculée comme suit :

$$\begin{aligned} \hat{S}_i^{lex} &= \text{Norm}(S_i^{lex}) \\ \hat{h}_{w_i} &= GRU_w(\hat{S}_i^{lex}, h_{w_{i-1}}) \\ h_{w_i} &= \text{FFNN}(\text{Norm}(\text{Dropout}(\hat{h}_{w_i}) + \hat{S}_i^{lex})) \end{aligned} \tag{7}$$

où nous avons indiqué avec Norm la normalisation des couches (*Layer Normalisation*) et avec *Dropout* la régularisation *dropout* (Srivastava *et al.*, 2014). Les autres couches GRU introduites plus haut sont modifiées de façon similaire.

Nous ne reprenons pas ici le mécanisme d'attentions multiples des *Transformer*, la prise en compte d'un contexte pertinent autour du mot à étiqueter reposant sur l'utilisation de la couche GRU_w (cf. section 2.1). Ce choix est motivé entre autre par les conclusions de Levy *et al.* (2018), qui montrent

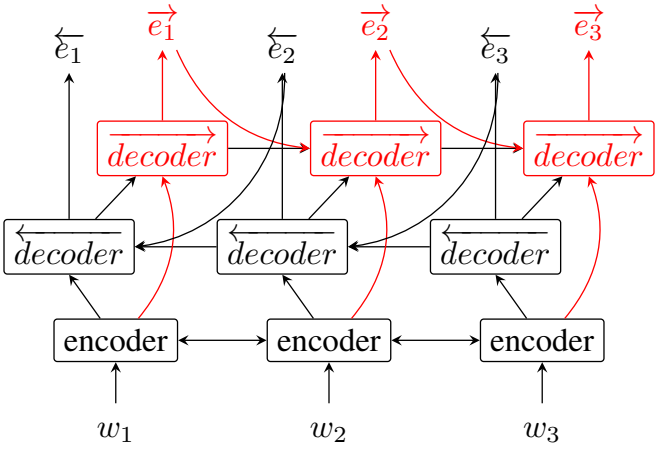

FIGURE 2 – Schéma de notre architecture neuronale, équivalente d'un point de vue global à celle de Dinarelli & Grobol (2019)

que les réseaux récurrents, dont GRU, encodent un long contexte sous forme de moyennes pondérée de toutes les entrées, constituant ainsi une forme d'attention. L'utilisation de couches GRU nous permet également de nous passer des *plongements positionnels* utilisés par Vaswani *et al.* (2017), la structure séquentielle des entrées étant implicitement encodée par les couches récurrentes GRU. [3]

L'architecture globale de notre réseau final peut être décrite par le même schéma utilisé que pour (Dinarelli & Grobol, 2019) (cf figure 2), les différences étant dans les calculs des couches décrits dans les sections précédentes.

3 Évaluation

Nous évaluons nos modèles sur trois tâches :

Le corpus français MEDIA (Bonneau-Maynard *et al.*, 2006) a été créé pour l'évaluation de systèmes de dialogues destinés à fournir des informations touristiques sur les hôtels en France. Les données ont été annotées manuellement suivant une ontologie de concepts riche. Des composants sémantiques peuvent être combinés pour former des étiquettes sémantiques complexes. [4] Les propriétés statistiques des données d'apprentissage, de développement et de test du corpus MEDIA sont données dans le tableau 1.

La tâche MEDIA peut être modélisée comme un étiquetage en concepts sémantiques de séquences en utilisant la convention *BIO* (Ramshaw & Marcus, 1995). Nous nous sommes déjà évalué sur cette tâche dans le passé, en utilisant une variété assez large de modèle probabilistes (Dinarelli *et al.*, 2009a,b ; Quarteroni *et al.*, 2009 ; Hahn *et al.*, 2010 ; Dinarelli, 2010 ; Dinarelli & Rosset, 2011 ; Dinarelli *et al.*, 2011).

Pour la tâche MEDIA des classes de mots sont disponibles pour permettre aux modèles une meilleurs généralisation sur certaines mots appartenant à des catégories dont des listes peuvent être facilement

3. Nous n'excluons pas cependant que l'utilisation des *plongements positionnels* puisse améliorer davantage notre architecture

4. Par exemple l'étiquette `localisation` peut être combinée avec les composants `ville`, `distance-relative`, `localisation-relative-générale`, `rue`, etc.

	Training		Validation		Test	
# phrases	12 908		1 259		3 005	
	Mots	Concepts	Mots	Concepts	Mots	Concepts
# Mots	94 466	43 078	10 849	4 705	25 606	11 383
dictionnaire	2 210	99	838	66	1 276	78
OOV%	–	–	1,33	0,02	1,39	0,04

TABLE 1 – Statistiques des données du corpus français MEDIA

	Training		Validation		Test	
# phrases	38 219		5 527		5 462	
	Mots	Étiquettes	Mots	Étiquettes	Mots	Étiquettes
# Mots	912 344	–	131 768	–	129 654	–
dictionnaire	43 210	45	15 081	45	13 968	45
OOV%	–	–	22,61	0	20,00	0

TABLE 2 – Statistiques des données du corpus anglais *WSJ*

récupérées dans le cadre d'une application d'interaction humain-machine comme les systèmes de dialogue. Des exemples de ces classes sont les noms des villes en France (classe *VILLE*), les noms des marques d'hôtel (*HOTEL*), les quantités correspondant à des montant (*MONTANT*), etc. Nous avons utilisés ces classes pour certaines expériences, ceci est indiqué par *FEAT* (pour *features*) dans les tableaux de la section suivante.

Le corpus anglais Penn TreeBank (Marcus *et al.*, 1993), indiqué avec WSJ dans la suite, constitue l'une des tâches les plus utilisées pour l'évaluation de modèles pour l'étiquetage de séquence. La tâche consiste à associer chaque mot avec son étiquette (*POS tag*). Nous utilisons la répartition habituelle des données : les sections 0-18 pour l'apprentissage, les sections 19-21 comme données de développement et les sections 22-24 comme données de test. Les propriétés statistiques des données d'apprentissage, de développement et de test du corpus WSJ sont données dans le tableau 2.

Le corpus allemand TIGER (Brants *et al.*, 2004) est annoté avec des informations morpho-syntaxiques riches, comprenant non seulement les *POS* comme dans le corpus WSJ, mais aussi le genre, le nombre, le cas et des informations de conjugaison pour les verbes. Cette tâche est proche du *POS tagging* d'un point de vue de modélisation, avec une plus grande difficulté liée non seulement à la langue, mais aussi au nombre assez grand d'étiquettes que le modèle doit désambiguïser (694 au total, contre 138 dans MEDIA et 45 dans le WSJ). Nous utilisons la même répartition de donnée que (Lavergne & Yvon, 2017). Les propriétés statistiques des données d'apprentissage, de développement et de test de ce corpus sont données dans le tableau 3. Comme nous pouvons le constater la taille des dictionnaires, que ce soit pour les mots ou pour les étiquettes, est assez importante.

Nous notons que le taux de mots hors vocabulaire (OOV) dans le corpus MEDIA est assez faible. Il y a en effet uniquement deux mots hors vocabulaire dans les données de développement et de test, il s'agit en plus de mots vides, il n'y a donc pratiquement pas de mots inconnus.

En revanche le taux de mots hors vocabulaire dans les corpus WSJ et TIGER est assez élevé : environ 1 sur 5 dans le premier et 1 sur 3 dans le second.

	Training		Validation		Test	
# phrases	40 472		5 000		5 000	
	Mots	Étiquettes	Mots	Étiquettes	Mots	Étiquettes
# Mots	719 530	–	76 704	–	92 004	–
dictionnaire	77 220	681	15 852	501	20 149	537
OOV%	–	–	30,90	0,01	37,18	0,015

TABLE 3 – Statistiques des données du corpus allemand TIGER

Modèle	**Précision**	**F1**	**CER**
MEDIA DEV			
GRU+LD-RNN	89.11	85.59	11.46
GRU+LD-RNN$_{le}$	89.42	86.09	10.58
GRU+LD-RNN$_{le}$ seg-len 15	**89.97**	**86.57**	**10.42**
fw-GRU+LD-RNN$_{le}$ seg-len 15	89.51	85.94	11.40

TABLE 4 – Comparaison des résultats sur les données de developpement du corpus MEDIA, sans et avec l'information lexicale au niveau des décodeurs $\overleftarrow{\text{GRU}}_e$ and $\overrightarrow{\text{GRU}}_e$ (Seq2Biseq$_{le}$ dans le tableau)

3.1 Réglages

Pour le développement de nos modèles nous avons utilisé le corpus MEDIA, le corpus plus petit et permettant donc une optimisation plus rapide. Nos réglages sont les mêmes que dans (Dinarelli *et al.*, 2017 ; Dinarelli & Grobol, 2019), qui utilisait également ces données pour l'évaluation. Les réglages sur le corpus WSJ sont les mêmes, sauf pour les plongements des mots (300 dimensions) et le taux d'apprentissage ($2{,}5 \times 10^{-4}$). Nous utilisons ces mêmes réglages pour WSJ et TIGER.

Comme nous l'avons expliqué dans (Dinarelli & Grobol, 2019), dans un premier temps nos modèles ne passaient pas dans la mémoire des nos GPU. Pour resoudre ce problème nous avons utilisé deux solutions. La première consiste en organiser les données d'apprentissage comme un seul flux de tokens. Ce flux est ensuite découpé en segments de taille fixe qui se chevauchent, avec un glissement d'un token entre deux segments consécutifs. La seconde solution est plus classique et consiste en regrouper ensemble les phrases de la même longueur. Ceci crée des groupes petits pour des phrases de grande taille, qui sont plus rares, et des grands groupes pour des phrases de taille petite et moyenne. Dans les deux cas les groupes passent en mémoire sans problème.

Dans nos expériences nous avons trouvé que la première solution marche bien mieux pour MEDIA, alors que sur les données *WSJ* et *TIGER* les deux solutions sont à peu près équivalentes. Pour des données de cette taille nous préférons donc la seconde solution, qui est plus intuitive et générale.

3.2 Résultats

Les résultats sur la tâche MEDIA sont des moyennes sur 10 expériences, les paramètres entraînables sont réinitialisés aléatoirement [5] pour chacune d'entre elles. Sur cette tâche nous nous évaluons en termes de précision et, puisqu'il faut reconstituer les concepts à partir des étiquettes BIO, aussi avec la F-mesure et le *Concept Error Rate* (CER). Le CER est calculé en alignant la prédiction avec l'annotation de référence, et en divisant ensuite la somme des insertions, substitutions et délétions par

5. Suivant une distribution uniforme pour les couches GRU et suivant (He *et al.*, 2015) pour les couches linéaires.

Modèle	Précision	F1	CER
MEDIA DEV			
GRU+LD-RNN	89.97	86.57	10.42
GRU+LD-RNN$_{2\text{-opt}}$	**90.22**	86.88	9.97
GRU+LD-RNN+FEAT$_{2\text{-opt}}$	90.14	**87.05**	**9.54**
MEDIA TEST			
BiGRU+CRF (Dinarelli *et al.*, 2017)	–	86.69	10.13
LD-RNN$_{\text{deep}}$ (Dinarelli *et al.*, 2017)	–	87.36	9.8
GRU+LD-RNN	89.57	87.50	10.26
GRU+LD-RNN$_{2\text{-opt}}$	89.79	87.69	9.93
GRU+LD-RNN+FEAT$_{2\text{-opt}}$	**90.12**	**87.94**	**9.48**

TABLE 5 – Performances des différentes variantes de notre architecture pour la tâche d'étiquetage sémantique sur MEDIA, comparées à l'état de l'art

le nombre de concepts de la référence. Sur les autres tâches, dans lesquelles à chaque mot correspond une étiquette, nous nous évaluons uniquement en termes de précision.

Les résultats présentés dans le tableau 4 sont les mêmes discutés dans notre précèdent travail (Dinarelli & Grobol, 2019), nous les ré-discutons également ici pour fournir un travail autonome et complet. Dans ces expériences nous avons voulu tester les capacités des décodeurs à construire une représentation efficace du contexte au niveau des étiquettes, et à filtrer des informations bruitées ou non-pertinentes dans la construction de telles représentations. Pour tester cela nous avons effectué des expériences sans (GRU+LD-RNN) et avec (GRU+LD-RNN$_{le}$) l'information lexicale h_{w_i} au niveau des décodeurs. Comme nous pouvons le constater dans le tableau 4 le modèle utilisant l'information lexicale obtient des résultats significativement meilleurs. Ceci, en prenant en compte que les deux modèles GRU+LD-RNN et GRU+LD-RNN$_{le}$ utilisent tous les deux l'information lexicale aussi au niveau de la couche de sortie (cf. section 2.3), prouve la capacité des décodeurs à construire une représentation plus efficace du contexte d'étiquettes en partant d'informations en entrée plus riches.

Pour tester la capacité à filtrer des informations non-pertinentes, nous avons effectué des expériences en variant la taille des segments de 10 (par défaut) à 15 tokens (cf section 3.1). À nouveau nous pouvons constater que les résultats s'améliorent avec des segments de taille 15 (GRU+LD-RNN$_{le}$ seg-len 15 dans le tableau 4). Compte tenu que MEDIA est constitué de transcriptions de l'orale, donc de données bruitées, ces améliorations montrent une bonne capacité de filtre des informations non-pertinentes par les décodeurs.

La dernière ligne du tableau 4 montre les résultats obtenus en n'utilisant que le décodeur *forward* (fw-GRU+LD-RNN$_{le}$ seg-len 15 dans le tableau). Ces résultats prouvent tout l'intérêt à utiliser à la fois un contexte gauche et un contexte droit au niveau des étiquettes, le modèle employant deux décodeurs étant largement meilleurs que celui en utilisant un seul.

Puisque l'utilisation de l'information lexicale h_{w_i} dans les décodeurs, et les segments de taille 15 mènent aux meilleurs résultats, ces réglages sont choisi par défaut et par la suite ils ne seront pas specifiés à côté du nom du modèle, qui sera simplement GRU+LD-RNN. Pour rappel, l'organisation des données en segment est utilisé uniquement pour MEDIA, pour les autres tâches nous utilisons une organisation en groupe de phrases de la même taille.

Modèle	**Précision**	
	WSJ DEV	WSJ TEST
LD-RNN$_{\text{deep}}$	96.90	96.91
LSTM+CRF (Ma & Hovy, 2016)	–	97.13
GRU+LD-RNN	97.13	97.20
GRU+LD-RNN$_{2-opt}$	**97.22**	**97.36**
LSTM+CRF + Glove (Ma & Hovy, 2016)	97.46	97.55
LSTM+LD-RNN + Glove (Zhang *et al.*, 2018)	–	97.59

TABLE 6 – Performances des différentes variantes de notre architecture pour la tâche de POS-tagging sur WSJ, comparées à l'état de l'art

Les résultats complet sur MEDIA sont donnés dans le tableau 5. Dans ce tableau nous montrons les résultats sur les données de développement et de test. Sur ces dernières nous nous comparons avec nos résultats précédents (Dinarelli *et al.*, 2017). Dans un premier temps nous avons entraîné un modèle sans les classes de mots disponibles pour cette tâche (GRU+LD-RNN, cf le début de la section 3), et sans utiliser les fonctionnalités d'un modèle *Transformer*. Alors que nous améliorions légèrement l'état-de-l'art en termes de F-mesure (87,50 contre 87,36), notre CER restait supérieur.

En analysant les sorties de nos modèles sur les données de développement nous avons remarqué des signes clairs indiquant que le modèle ignorait l'information donnée par le contexte droit au niveau des étiquettes. Nous en avons conclus que notre modèle souffrait du même problème mentionné dans (Vaswani *et al.*, 2017) pour les RNNs. Nous notons que ce problème, bien qu'il a été remarqué sur des données particulières, concerne une limitation du modèle sur sa capacité à prendre en compte le contexte droit, il s'agit donc d'un comportement générale et non lié à ce jeu de données spécifiques. Comme nous le verrons pas la suite, en effet les modifications mises en place pour résoudre ce problème se sont avérées assez efficaces sur toutes les tâches sur lesquelles nous nous évaluons.

Nous avons alors ajouté à notre architecture les fonctionnalités du modèle *Transformer* mentionnées dans la section 2.5. Puisque notre modèle utilise deux décodeurs, nous avons aussi entraîné le système avec 2 optimiseurs (GRU+LD-RNN$_{\text{2-opt}}$), chacun minimisant la log-vraisemblance de la sortie de chaque décodeur. Les résultats obtenus avec ce modèle dépassent tous les précédents pour toutes les mesures d'évaluation, et dépassent également l'état-de-l'art dans les mêmes conditions.

En ajoutant les classes des mots disponibles dans MEDIA (GRU+LD-RNN+FEAT$_{\text{2-opt}}$), les résultats s'améliorent encore pour toutes les mesures d'évaluation.

Étant donné leur taille réduite, nous avons utilisé les données MEDIA pour une optimisation plus rapide des choix au niveau de l'architecture et de la plupart des hyper-paramètres. Les seuls paramètres que nous avons ré-optimisé sur le corpus WSJ sont le taux d'apprentissage et la taille de la couche cachée (cf. section 3.1). La taille des plongements des mots a été choisie en se basant sur (Zhang *et al.*, 2018). Nous avons effectué une seule expérience sur le corpus TIGER. Puisque celle-ci a donné des bons résultats, nous n'avons pas optimisé davantage le modèle.

Les résultats sur le *POS tagging* du corpus WSJ sont montrés dans le tableau 6. Dans ce cas également l'utilisation des fonctionnalités d'un modèle *Transformer* donne des améliorations sur la précision. Nous notons que notre modèle, que ce soit avec un seul optimiseur ou deux et avec en plus les fonctionnalités du *Transformer*, améliore le modèle *LSTM+CRF* (Ma & Hovy, 2016) sans utiliser des

Modèle	**Précision**	
	TIGER DEV	TIGER TEST
GRU+LD-RNN$_{2-opt}$	**93.90** (98.30)	**91.86** (97.74)
VO-CRF (Lavergne & Yvon, 2017)	–	88.78

TABLE 7 – Performances des différentes variantes de notre architecture pour l'étiquetage morpho-syntaxique sur TIGER, comparées à l'état de l'art

plongements pré-appris avec *GloVe* (Pennington *et al.*, 2014). À titre de comparaison nous montrons aussi les meilleurs résultats de la littérature sur cette tâche. Bien que nos résultats ne dépassent pas l'état-de-l'art, ils en restent assez proche. Nous notons cependant que nos premières expériences avec des plongements pré-appris n'améliorent pas l'état-de-l'art [6]. Des analyses sont en cours pour comprendre ce manque de gain.

Dans le tableau 7 nous montrons les résultats obtenus sur la tâche d'étiquetage morpho-syntaxique de l'allemand (TIGER). À notre connaissance le meilleur résultat de la littérature est celui obtenu par Lavergne & Yvon (2017) avec un CRF d'ordre variable (VO-CRF). Nous pouvons donc constater que notre modèle améliore l'état-de-l'art sur cette tâche. Entre parenthèses nous montrons également les résultats du *POS tagging*. Ces résultats sont obtenus des précédents en ne considérant que l'étiquette POS, sans apprentissage spécifique.

4 Conclusions

Dans cet article nous avons proposé un modèle neuronal pour la modélisation de séquences qui réunit des caractéristiques des modèles neuronaux plus populaires de ces dernières années : les RNNs bi-directionnels, les modèles *encodeur-décodeur* et le modèle *Transformer*. Une évaluation sur trois tâches classiques d'étiquetage de séquences montre que notre modèle est très efficace pour ce type de problèmes. En effet il obtient souvent des résultats à l'état-de-l'art, et il en est proche dans tous les cas.

Remerciements

Cette recherche s'insère dans le programme « Investissements d'Avenir » géré par l'Agence Nationale de la Recherche ANR-10-LABX-0083 (Labex EFL).

Ce travail a par ailleurs bénéficié du soutien de l'ANR DEMOCRAT (Description et modélisation des chaînes de référence : outils pour l'annotation de corpus et le traitement automatique), projet ANR-15-CE38-0008.

6. Contrairement au modèle *LSTM+CRF* (Ma & Hovy, 2016) dont la précision est grandement améliorée par les plongements pré-appris *GloVe*.

Références

Bahdanau D., Cho K. & Bengio Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, **abs/1409.0473**.

Bonneau-Maynard H., Ayache C., Bechet F., Denis A., Kuhn A., Lefèvre F., Mostefa D., Qugnard M., Rosset S. & Servan, S. Vilaneau J. (2006). Results of the french evalda-media evaluation campaign for literal understanding. In *LREC*, p. 2054–2059, Genoa, Italy.

Brants S., Dipper S., Eisenberg P., Hansen-Schirra S., Konig E., Lezius W., Rohrer C., Smith G. & Uszkoreit H. (2004). TIGER : Linguistic interpretation of a german corpus. *Research on Language and Computation*, **2**(4), 597–620.

Chen M. X., Firat O., Bapna A., Johnson M., Macherey W., Foster G., Jones L., Schuster M., Shazeer N., Parmar N., Vaswani A., Uszkoreit J., Kaiser L., Chen Z., Wu Y. & Hughes M. (2018). The Best of Both Worlds : Combining Recent Advances in Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, p. 76–86 : Association for Computational Linguistics.

Cho K., van Merrienboer 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 : Association for Computational Linguistics.

Collins M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL*, p. 16–23, Stroudsburg, PA, USA : Association for Computational Linguistics.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K. & Kuksa P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, **12**.

Dai Z., Yang Z., Yang Y., Carbonell J., Le Q. V. & Salakhutdinov R. (2019). Transformer-XL : Attentive Language Models Beyond a Fixed-Length Context. *arXiv preprint 1901.02860*.

De Mori R., Bechet F., Hakkani-Tur D., McTear M., Riccardi G. & Tur G. (2008). Spoken language understanding : A survey. *IEEE Signal Processing Magazine*, **25**, 50–58.

Dehghani M., Gouws S., Vinyals O., Uszkoreit J. & Kaiser L. (2018). Universal transformers. *CoRR*, **abs/1807.03819**.

Devlin J., Chang M.-W., Lee K. & Toutanova K. (2018). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint 1810.04805*.

Dinarelli M. (2010). *Spoken Language Understanding : from Spoken Utterances to Semantic Structures*. PhD thesis, International Doctoral School in Information and Communication Technology, Dipartimento di Ingegneria e Scienza dell' Informazione, via Sommarive 14, 38100 Povo di Trento (TN), Italy.

Dinarelli M. & Grobol L. (2019). Seq2biseq : Bidirectional output-wise recurrent neural networks for sequence modelling. *CoRR*, **abs/1904.04733**.

Dinarelli M., Moschitti A. & Riccardi G. (2009a). Concept segmentation and labeling for conversational speech. In *Proceedings of the International Conference of the Speech Communication Assosiation (Interspeech)*, Brighton, U.K.

Dinarelli M., Moschitti A. & Riccardi G. (2009b). Re-ranking models based on small training data for spoken language understanding. In *Conference of Empirical Methods for Natural Language Processing*, p. 11–18, Singapore.

Dinarelli M., Moschitti A. & Riccardi G. (2011). Discriminative reranking for spoken language understanding. *IEEE TASLP*, **20**, 526–539.

Dinarelli M. & Rosset S. (2011). Hypotheses selection criteria in a reranking framework for spoken language understanding. In *Conference of Empirical Methods for Natural Language Processing*, p. 1104–1115, Edinburgh, U.K.

Dinarelli M. & Rosset S. (2012a). Tree representations in probabilistic models for extended named entity detection. In *European Chapter of the Association for Computational Linguistics (EACL)*, p. 174–184, Avignon, France.

Dinarelli M. & Rosset S. (2012b). Tree-structured named entity recognition on ocr data : Analysis, processing and results. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk & S. Piperidis, Eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey : European Language Resources Association (ELRA).

Dinarelli M. & Tellier I. (2016a). Improving recurrent neural networks for sequence labelling. *CoRR*, **abs/1606.02555**.

Dinarelli M. & Tellier I. (2016b). New recurrent neural network variants for sequence labeling. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics*, Konya, Turkey : Lecture Notes in Computer Science (Springer).

Dinarelli M., Vukotic V. & Raymond C. (2017). Label-dependency coding in Simple Recurrent Networks for Spoken Language Understanding. In *Interspeech*, Stockholm, Sweden.

Dupont Y., Dinarelli M. & Tellier I. (2017). Label-dependencies aware recurrent neural networks. In *Proceedings of CICling*, Budapest, Hungary : LNCS, Springer.

Grouin C., Dinarelli M., Rosset S., Wisniewski G. & Zweigenbaum P. (2011). Coreference resolution in clinical reports. the limsi participation in the i2b2/va 2011 challenge. In *In Proceedings of i2b2/VA 2011 Coreference Resolution Workshop*.

Guo Q., Qiu X., Liu P., Shao Y., Xue X. & Zhang Z. (2019). Star-transformer. *CoRR*, **abs/1902.09113**.

Hahn S., Dinarelli M., Raymond C., Lefèvre F., Lehen P., De Mori R., Moschitti A., Ney H. & Riccardi G. (2010). Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE TASLP*, **99**.

He K., Zhang X., Ren S. & Sun J. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, p. 1026–1034.

Lample G., Ballesteros M., Subramanian S., Kawakami K. & Dyer C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 260–270 : Association for Computational Linguistics.

Lavergne T. & Yvon F. (2017). Learning the structure of variable-order crfs : a finite-state perspective. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 433–439 : Association for Computational Linguistics.

Levy O., Lee K., FitzGerald N. & Zettlemoyer L. (2018). Long short-term memory as a dynamically computed element-wise weighted sum. In *Proceedings of ACL*, p. 732–739 : ACL.

Ma X. & Hovy E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of ACL*.

Marcus M. P., Santorini B. & Marcinkiewicz M. A. (1993). Building a large annotated corpus of english : The penn treebank. *COMPUTATIONAL LINGUISTICS*, **19**(2).

Ng V. & Cardie C. (2002). Improving Machine Learning Approcahes to Corefrence Resolution. In *Proceedings of ACL'02*, p. 104–111.

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., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. & Zettlemoyer L. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, volume 1, p. 2227–2237 : Association for Computational Linguistics.

Quarteroni S., Riccardi G. & Dinarelli M. (2009). What's in an ontology for spoken language understanding. In *Proceedings of the International Conference of the Speech Communication Assosiation (Interspeech)*, Brighton, U.K.

Ramshaw L. & Marcus M. (1995). Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, p. 84–94, Cambridge, MA, USA.

Rush A. M., Reichart R., Collins M. & Globerson A. (2012). Improved parsing and pos tagging using inter-sentence consistency constraints. In *Proceedings of EMNLP-CoNLL*, Stroudsburg, PA, USA.

Soon W. M., Ng H. T. & Lim D. C. Y. (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, **27**(4), 521–544.

Srivastava N., Hinton G., Krizhevsky A., Sutskever I. & Salakhutdinov R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**, 1929–1958.

Sutskever I., Vinyals O. & Le Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, Cambridge, MA, USA : MIT Press.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L. & Polosukhin I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett, Eds., *Advances in Neural Information Processing Systems 30*, p. 5998–6008. Curran Associates, Inc.

Vinyals O., Kaiser L., Koo T., Petrov S., Sutskever I. & Hinton G. (2015). Grammar As a Foreign Language. In *Proceedings of the 28th International Conference on Neural Information Processing*, volume 2 of *NIPS'15*, p. 2773–2781, Cambridge, MA, USA : MIT Press.

Zhang Y., Chen H., Zhao Y., Liu Q. & Yin D. (2018). Learning tag dependencies for sequence tagging. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

PolylexFLE : une base de données d'expressions polylexicales pour le FLE

Amalia Todirascu[1] Marion Cargill[1] Thomas François[2]
(1) LiLPa, Université de Strasbourg, 22, rue René Descartes, 67084 Strasbourg, France
(2) CENTAL, UCLouvain, Place Montesquieu, 3, bte L2.03.02, 1348 Louvain-la-Neuve, Belgique
todiras@unistra.fr, mcargill@unistra.fr, thomas.francois@uclouvain.be

RÉSUMÉ

Nous présentons la base PolylexFLE, contenant 4295 expressions polylexicales. Elle est integrée dans une plateforme d'apprentissage du FLE, SimpleApprenant, destinée à l'apprentissage des expressions polylexicales verbales (idiomatiques, collocations ou expressions figées). Afin de proposer des exercices adaptés au niveau du Cadre européen de référence pour les langues (CECR), nous avons utilisé une procédure mixte (manuelle et automatique) pour annoter 1098 expressions selon les niveaux de compétence du CECR. L'article se concentre sur la procédure automatique qui identifie, dans un premier temps, les expressions de la base PolylexFLE dans un corpus à l'aide d'un système à base d'expressions régulières. Dans un second temps, leur distribution au sein de corpus, annoté selon l'échelle du CECR, est estimée et transformée en un niveau CECR unique.

ABSTRACT

PolylexFLE : a database of multiword expressions for French L2 language learning

We present the PolylexFLE database, containing 4295 polylexical expressions. It is integrated into a platform to support learning of verbal polylexical expressions (idioms, collocations or fixed expressions). In order to propose exercises adapted to the level of the European Framework of Reference for Languages (CEFR), we used a mixed approach (manual and automatic) to annotate 1098 expressions according to the CEFR levels. The paper focuses on the automatic procedure that first identifies the expressions from the PolylexFLE database in a corpus using a regular expression-based system. In a second step, their distribution in this corpus, labelled according to the CEFR scale, is estimated and transformed into a single CEFR level.

MOTS-CLÉS : expressions polylexicales vebales, niveau CECR, TAL pour la didactique du FLE.

KEYWORDS: verbal multiword expressions, CEFR level, NLP for French L2 language learning.

1 Contexte et motivation

Les expressions polylexicales (EP) constituent une classe d'objets linguistiques qui inclut les expressions idiomatiques, les expressions figées et des collocations. La définition exacte des EP reste discutée, mais plusieurs chercheurs s'accordent à les identifier comme "des séquences de mots, dont le sens est plus ou moins compositionnel, caractérisés par des propriétés morpho-syntaxiques, syntaxiques, sémantiques" (Baldwin & Kim, 2010). Ces unités seraient stockées directement en mémoire, ce qui rend leur traitement cognitif plus rapide (Pawley & Syder, 1983), du moins dans le cas de natifs. Les apprenants d'une langue étrangère éprouvent, quant à eux, bien des difficultés

dans l'acquisition et le traitement des EP. Leur maitrise des EP se situe souvent bien en-deçà de leurs connaissances lexicales générales (Bahns & Eldaw, 1993) et ils tendent à effectuer des traductions mot à mot de ces expressions, ignorant le sens figuré de ces expressions. De récentes études montrent pourtant qu'une bonne maitrise des EP améliore la compréhension en lecture (Kremmel *et al.*, 2017).

À ces difficultés d'acquisition vient s'ajouter le fait que les plateformes en ligne proposent rarement des exercices visant directement l'apprentissage des EP. Pour le français langue étrangère (FLE), *Bonjour de France* et *Le point du FLE* constituent à cet égard des exceptions. Autre exemple, une plateforme comme *The Writing Mentor* [1] qui vise à soutenir le développement des compétences de production écrite via des annotations collaboratives et un apprentissage par feedback (Hamel *et al.*, 2016), ne se focalise pas sur les expressions polylexicales. Quant aux plateformes de création d'activités pour l'apprentissage des langues [2], celles-ci se concentrent seulement sur certaines facettes du vocabulaire (élargissement par synonymie, reformulations), mais les EP ne font que rarement l'objet de corrections ou de retours proposés à l'utilisateur. Toutefois, *Language Muse* propose des mots composés ou des verbes à particules en anglais pour améliorer la lecture des apprenants (Madnani *et al.*, 2016). C'est pourquoi, nous avons cherché à développer une plateforme dédiée à la problématique des EP dans le cadre du projet SimpleApprenant, qui est décrit à la section 2.

Dans ce cadre, un problème nous est rapidement apparu : la rareté des ressources proposant une base d'EP adaptée à un usage pédagogique. En effet, afin de proposer des exercices adaptés au niveau des apprenants, il convient de disposer d'une base de données dans laquelle le niveau de difficulté des EP est signalé, de préférence en rapport avec l'échelle du Cadre européen commun de référence pour les langues (CECR), publié par le Conseil de l'Europe (2001) afin de structurer le secteur de l'enseignement des langues étrangères au niveau européen. Or, il existe également très peu de plateformes d'apprentissage où les exercices et les ressources portant sur les EP sont annotées selon le niveau CECR.

Plusieurs ressources sont disponibles pour l'apprentissage des expressions idiomatiques et des collocations. Ainsi, la *Base Lexicale du Français* (Verlinde *et al.*, 2006) propose une description détaillée des contextes syntaxiques et morpho-syntaxiques des EP, des traductions et des exemples tirés de corpus. Le projet *DIRE Autrement* (Hamel & Milicevic, 2007) propose quant à lui les collocatifs les plus fréquents pour certaines expressions et des exercices permettant de mettre en correspondance des expressions et des définitions. Le projet PARSEME-FR a crée un corpus annoté en EP (Ramisch *et al.*, 2018), mais adopte une classification très détaillée des expressions. Ces travaux ne font toutefois pas de liens entre les EP envisagées et l'échelle du CECR.

Plus proche de ce qui nous intèresse, *EmoProf*, intégré dans la base lexicale *EmoBase* (Diwersy *et al.*, 2014), propose des séquences didactiques pour les professeurs de FLE ciblant des expressions polylexicales liées au vocabulaire des émotions (verbes, noms, adjectifs). Les séquences didactiques sont classées par niveau (Cavalla *et al.*, 2013), mais ces séquences ne sont pas directement exploitables par une application de TAL. De même, il existe les référentiels pour le FLE de Beacco *et al.* (2004) qui incluent des EP dans les descriptions lexicales des niveaux du CECR. À nouveau, le format (papier) de cette ressource n'est pas exploitable dans un contexte de TAL. Il existe bien la ressource FLELex (François *et al.*, 2014) qui précise, pour plus de 2000 EP, leur distribution sur les 6 niveaux du CECR. Celle-ci est directement exploitable pour le TAL, mais intègre surtout des EP nominales. Au niveau des EP verbales, pourtant cruciales dans le contexte de l'apprentissage du FLE, on ne trouve dans FLELex que des formes à base des verbes *faire* (ex. *faire part*, *faire obstacle*) et *avoir* (ex.

1. https://mentormywriting.org/
2. https://languagemuse.org/

avoir peur, *avoir faim*).

Dans cet article, nous proposons une nouvelle base de données pédagogique pour les expressions polylexicales, PolylexFLE, dédiée aux EP verbales. Elle est intégrée dans la plateforme SimpleApprenant dont les exercices sont majoritairement dédiés à l'apprentissage des EP. Nous présentons tout d'abord cette plateforme à la section 2, avant de décrire le contenu de PolylexFLE et le processus de collecte des EP à la section 3. Ensuite, nous détaillons à la section 4 la façon dont un niveau de compétence du CECR a pu être associé à 1098 EP à l'aide d'une approche mixte : manuelle (basée sur des vocabulaires de référence) et automatique (basée sur le traitement automatique de corpus pédagogiques). Par rapport aux travaux existants, notre base PolylexFLE présente deux avantages : les EP y sont annotées avec leur niveau CECR et le format est directement utilisable pour le TAL.

2 Le projet SimpleApprenant

La base de données PolylexFLE a été développée dans le cadre du projet SimpleApprenant [3], qui a pour objectif d'aider les apprenants du FLE à améliorer leurs compétences écrites et leur connaissance des EP (Todirascu & Cargill, 2019). Dans ce but, une application web qui repose sur des outils et des ressources TAL a été développée, proposant 3 fonctionnalités :

— améliorer les connaissances des EP à travers des exercices, étalonnés en fonction du niveau CECR.
— suggérer des EP en relation avec un mot introduit par un utilisateur.
— corriger automatiquement au niveau typographique, lexical ou syntaxique, un texte écrit par un apprenant, à l'aide d'un module TAL, SimplifyYourFrench.

Comme la plateforme dispose des profils des utilisateurs, qui renseignent entre autres leur niveau CECR, elle peut leur proposer des exercices et des EP adaptés à leur niveau. Pour ce faire, il a cependant été nécessaire de mettre au point une base d'EP, PolylexFLE, qui renseigne le niveau CECR de chaque EP. Nous détaillons, à la section suivante, la méthode de conception de cette base lexicale.

3 La conception de PolylexFLE

La base PolylexFLE comporte, dans son ensemble, 4295 entrées (sous forme de lemmes), associées à leurs patrons syntaxiques, thématiques (par exemple : couleurs, parties du corps), traductions, définitions (extraites automatiquement du Wiktionnaire) et phrases en contexte. Parmi toutes ces entrées, 1098 EP sont également associées à un niveau CECR.

Les 4295 expressions présentes dans la base ont été extraites du Lexique-Grammaire (Gross, 1994; Laporte *et al.*, 2008), ressource lexicale présentant des verbes et des EP verbales associées à leurs informations syntaxiques, morpho-syntaxiques et sémantiques. Nous avons sélectionné des expressions et leurs contextes syntaxiques, sous forme de patrons syntaxiques. Nous avons filtré ces expressions selon les définitions proposées par (Baldwin & Kim, 2010). Ainsi, nous intégrons dans la base des expressions qui sont composées d'au moins deux unités lexicales, reliées par des dépendances syntaxiques, qui présentent au moins des spécificités syntaxiques, sémantiques ou morphologiques (Constant *et al.*, 2017). De ce fait, nous avons identifié trois catégories d'EP :

3. https ://simpleapprenant.huma-num.fr/SimplifyYourFrench/accueil.jsp

— les expressions idiomatiques, qui ont un sens figuré, non déductible à partir des sens de chaque unité : ex. *perdre pied* (perdre la confiance en soi), *jeter l'éponge* (abandonner). Ces expressions sont caractérisées par l'absence du déterminant pour le nom (*perdre pied*, mais pas **perdre les pieds*), ou la préférence pour un déterminant précis (*faire le clown* mais pas **faire les clowns*), l'impossibilité de modifier le nom (**perdre pied gauche*) ou l'impossibilité d'utiliser un adverbe entre le verbe et le nom (**avoir toujours d'autres chats à fouetter*) et l'impossibilité de passivation. Ces expressions posent le plus de problèmes aux apprenants, car il est parfois difficile de trouver la traduction correcte ou une expression équivalente.
— les collocations manifestent une préférence lexicale forte (*poser une question*, mais pas **demander une question*), dans un champ lexical restreint (*hisser le pavillon/le drapeau*), leur sens est plutôt compositionnel. La variabilité syntaxique de ce type d'EP est importante : la modification du verbe est possible (*prendre rapidement des mesures* ; la modification du nom également (*prendre des mesures drastiques*) ; enfin le passage à la diathèse passive est acceptable. Les éléments de la collocation résistent aux tests de substitution (remplacer le nom ou le verbe par un synonyme). Signalons que notre définition de la classe des collocations est différente de celle adoptée par le projet PARSEME-FR pour la campagne d'annotation des expressions verbales (Ramisch *et al.*, 2018) où les collocations sont des mots qui cooccurrent fréquemment (*lire un livre/un article/des PDFs/une carte/des BDs/un fichier*).
— les expressions figées, comprenant les expressions dont le verbe est conjugué (*être sans espoir pour*, *avoir droit à*, *être d'accord*), mais dont l'objet est totalement fixe et lexicalisé (le déterminant est fixe et le nom n'accepte pas des modifications morphologiques, des adjectifs ne peuvent pas se combiner avec le nom). Dans la catégorie des expressions figées, nous incluons les expressions ayant une valeur pragmatique, étiquétées comme « pragmatème » (Tutin *et al.*, 2015) : *ça va s'arranger, le soleil brille, il fait chaud*. Pour ces expressions, on ne peut pas insérer un déterminant, un adjectif ou une relative. Gross (1993) parle de groupe nominal figé, d'adverbe figé, etc. On inclut la préposition si elle indique un des arguments de l'expression. Ces expressions représentent des difficultés pour un apprenant du FLE, car une partie est fixe et les contraintes syntaxiques et morpho-syntaxiques sont importantes.

4 Identification des niveaux de difficulté

Les expressions que nous avons extraites du Lexique-Grammaire ne sont pas annotées en niveaux CECR. Pour ajouter cette information à notre liste d'expressions, nous avons utilisé une approche mixte : manuelle, dans laquelle les niveaux sont obtenus à partir de ressources pédagoqiques et automatique, qui utilise des techniques de TAL.

L'approche manuelle est facile à décrire. Nous avons consulté des vocabulaires de référence pour l'apprentissage du FLE listant des expressions ainsi que, pour chacune de ces expressions, le niveau CECR auquel elle est supposée apprise. Il s'agit des référentiels de Beacco *et al.* (2004, 2008), ainsi que de manuels de FLE (Rey, 2007). Nous avons ainsi retrouvé 535 de nos expressions dans ces sources et leur avons attribué le même niveau CECR que celui renseigné dans ces sources. Nous avons renseigné ces niveaux dans la base.

Toutefois, une majorité de nos EP n'étaient pas reprises dans ces sources et nous avons dû mettre au point une méthodologie empirique basée sur corpus afin de leur attribuer automatiquement un niveau de difficulté. Cette technique consiste à identifier ces EP dans un corpus de textes (décrit à la

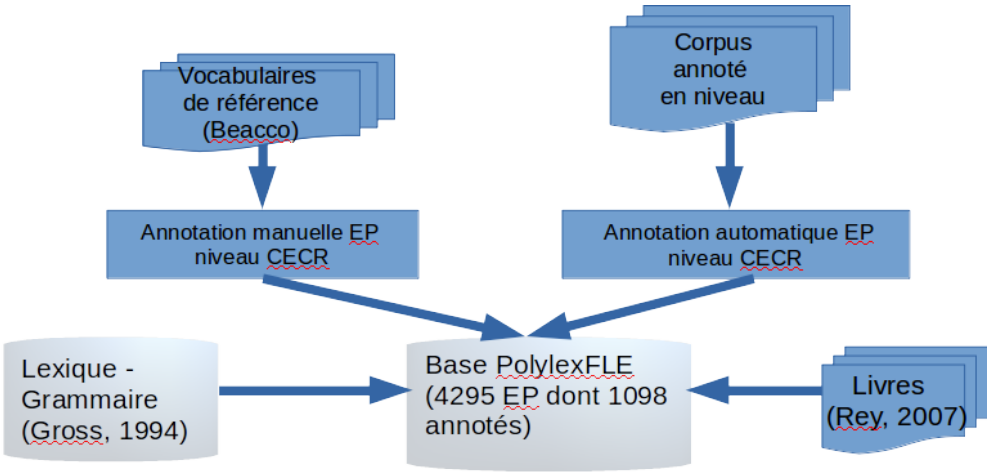

FIGURE 1 – Méthodologie d'annotation du niveau CECR

section 4.1) de FLE dont le niveau CECR est connu, ce qui permet de calculer leur distribution de fréquence par niveau du CECR. Ces étapes sont respectivement décrites aux sections 4.2 et 4.4, tandis que la qualité de l'identification automatique des expressions est évaluée à la section 4.3. Une fois ces distributions de fréquence obtenues, celles-ci sont transformées en un niveau unique (*cf.* section 4.4). Si le niveau de l'EP n'est pas renseigné dans la base, ce niveau unique calculé automatiquement sera ajouté. Si l'expression a déjà un niveau annoté manuellement, nous gardons ce niveau. Nous avons également effectué une évaluation préliminaire de la fiabilité de nos annotations CECR auprès d'apprenants du FLE, dont les résultats sont décrits à la section 5.

4.1 Description du corpus

Le corpus utilisé regroupe une quantité notable de textes extraits de manuels utilisés pour enseigner le FLE et associés à un niveau CECR. Ainsi, chaque texte se voit attribuer le même niveau du CECR que le manuel ou la ressource pédagogique dont il provient. Nous avons rassemblé plusieurs textes, regroupés en deux sous-corpus dont les caractéristiques divergent quelque peu. Le *corpus1*, collecté dans le cadre de cette étude, reprend des romans ou des contes très courts s'adressant aux apprenants du FLE. Il s'agit de textes écrits directement pour les apprenants, qui sont aussi accompagnés d'un lexique expliquant les mots et les termes les plus compliqués. Le *corpus2* correspond quant à lui au corpus décrit en détails dans François (2014). Il s'agit de textes directement extraits de manuels de FLE généralistes, destinés à des adultes ou grands adolescents et publiés après 2001. Le corpus comprend différents genres de textes, notamment des textes narratifs, informatifs, dialogiques et argumentatifs, mais aussi des petites annonces, des poèmes, des chansons, des recettes de cuisine, etc. La taille de chaque composante de ces deux corpus est reprise à la table 1. Pour notre étude, ces deux corpus ont été rassemblés au sein d'un corpus unique appelé *Corpus-M*, qui comprend 857 737 mots.

	A1	A2	B1	B2	C1	C2	Total
Corpus1	15 620	43 422	57 795	101 361	54 057	52 290	324 545
Corpus2	62 592	95 117	176 973	71 701	92 327	34 482	533 192
Corpus-M	78 212	138 539	234 768	173 062	146 384	86 772	857 737

TABLE 1 – Taille du corpus (en nombre de mots) par niveau du CECR

Pour réaliser l'annotation des EP selon l'échelle du CECR, nous examinons la distribution des EP dans ces deux corpus. Nous faisons l'hypothèse que l'expression trouvée dans un niveau CECR doit être connue par les apprenants ayant acquis ce niveau. Cela requiert tout d'abord d'être capable de détecter automatiquement ces EP verbales, ce qui constitue un sérieux défi, même au sein du domaine de l'extraction automatique d'EP. La section suivante détaille la technique mise au point dans ce but.

4.2 Méthode d'extraction des EP

Notre définition des EP répose sur les notions d'idiomaticité syntaxique, sémantique et morpho-syntaxique, selon Baldwin & Kim (2010) et Constant *et al.* (2017) : en plus des combinaisons statistiques fréquentes, les EP sont caractérisées par des liens syntaxiques et sémantiques et par des fortes préférences lexico-syntaxiques.

Nous annotons le corpus à l'aide de l'analyseur syntaxique Mind the Gap (Coavoux & Crabbé, 2017) et de Glàff (Hathout *et al.*, 2014). Ainsi, Mind the Gap applique une annotation syntaxique en dépendances que l'on peut exploiter pour extraire les expressions à partir du corpus. Cet analyseur identifie aussi certaines EP (en particulier, les locutions prépositionnelles, les noms propres), mais pas les EP verbales. Mind the Gap fait une analyse morphosyntaxique détaillée sans identifier les lemmes ; ceux-ci sont ajoutés à l'aide du dictionnaire Glàff sur la base de la catégorie lexicale identifiée par Mind the Gap.

L'identification des EP au sein du corpus est une tâche difficile, car les EP ne sont pas toujours rencontrées dans les textes sous la même forme que celle indexée dans la base. Notre approche est symbolique, utilisant les annotations morpho-syntaxiques, la lemmatisation, les dépendances syntaxiques et les patrons stockés dans la base. Parmi les trois catégories d'expressions (*cf.* section 3), les expressions idiomatiques et les expressions figées acceptent très peu de variations. Généralement, seul l'ajout d'un adverbe est possible : *il fait chaud/il fait très chaud* ; *il a jeté l'éponge/ il a jeté rapidement l'éponge*. Toutefois, les verbes qui composent les expressions peuvent se conjuger avec un auxiliaire. Les collocations, quant à elles, sont caractérisées par une grande variabilité syntaxique : variation du déterminant, insertion de modifieurs pour le nom ou pour le verbe, passivation, nominalisation.

Avant d'effectuer la recherche des expressions dans le corpus, nous générons toutes les variantes possibles pour chaque collocation, à l'aide des patrons syntaxiques provenant du Lexique Grammaire, qui sont représentés dans la base. Par exemple, pour la collocation *garder le silence*, nous utilisons le patron du Lexique-Grammaire associé :

<ENT>V_<ENT>Det1_<ENT>C1, ou V est le verbe (garder), Det1 est le déterminant (le) et C1 est le nom (silence), <ENT> est une balise qui sépare les éléments du patron.

A partir de ce patron, on génère les variantes possibles, en rajoutant des modifieurs adjectivaux (<ENT>V_<ENT>Det1_<ENT>C1_<ENT>Adj1), les modifieurs adverbiaux (<ENT>V_<ENT>Adv_<ENT>Det1_<ENT>C1) (*garder prudemment le silence*) ou les patrons qui indiquent la passivation (<ENT>Det1_<ENT>C1_<ENT>_être<ENT>VPP, VPP est le verbe au participe passé) (*le silence a été gardé*).

Parmi les variantes possibles, nous identifions également les nominalisations (*mise à jour*) ou les noms modifiés par un participe passé du verbe (*le silence bien gardé*). Pour les autres catégories d'expressions polylexicales, telles que les expressions idiomatiques et les expressions figées, nous ajoutons les auxiliaires (*j'ai eu d'autres chats à fouetter*) et les modifieurs adverbiaux (*il faisait*

tellement beau). Au terme de cette étape, ces variantes ont été ajoutées aux EP listées dans la base. La recherche peut alors être effectuée sur le corpus complet.

Ces patrons sont implémentés dans un système à base de règles, en Perl, utilisant des expressions régulières qui génèrent les variantes morpho-syntaxiques sur la base des patrons du Lexique Grammaire pour chaque candidat de la base. Ainsi, l'algorithme lit le texte phrase par phrase. Dès qu'un candidat est identifié dans la phrase courante, on vérifie les contraintes morpho-syntaxiques et syntaxiques associées dans la base. Ainsi, on vérifie les relations de dépendandance entre le nom et le verbe (complément d'objet direct ou indirect, sujet), la distance entre le verbe et le nom (limitée à 4 mots). De plus, on vérifie la catégorie lexicale des mots insérés entre le nom et le verbe (adjectifs ou adverbes, les éventuels determinants qui précedent le nom).

Évaluation. Des corpus annotés en EP sont disponibles, mais nous avons voulu évaluer la qualité de notre extraction directement sur des documents de FLE, dont les caractéristiques sont assez éloignées des textes utilisés dans les corpus d'évaluation classique. Nous avons dès lors sélectionné aléatoirement 20 textes représentatifs des niveaux A1 à C2 (A1 : 366 mots ; A2 : 724 mots ; B1 : 2 059 mots ; B2 : 2 008 mots ; C1 : 2 425 mots ; C2 : 453 mots) dans le *Corpus2*.

L'annotation des EP verbales de ce corpus d'évaluation a été réalisée à l'aide d'un guide d'annotation. Notre guide d'annotation diffère du guide d'annotation proposé dans le cadre de la campagne d'annotation des EP verbales par le projet PARSEME [4]. En effet, PARSEME propose une classification très détaillée des EP verbales, valable pour un grand nombre de langues : construction à verbes supports (*faire peur*), construction à verbes multiples (*to let go*), verbes à particules (*give up*), etc. Pour le français, les ressources et les outils créés dans le cadre du projet PARSEME FR permettent l'annotation des expressions polylexicales spécifiques. Ainsi, ces outils annotent les formes verbales pronominales (*se laver, s'assoir*), les expressions idiomatiques et des constructions à verbe support (*faire peur, prendre une décision*). Pour notre application, qui s'adresse aux apprenants de FLE, la classification est restreinte aux trois catégories présentées à la section 3 : expressions idiomatiques, collocations et expressions figées. Notre classification coincïde avec la classification proposée par PARSEME-FR en ce qui concerne les expressions idiomatiques. Les constructions à verbes supports peuvent être retrouvées parmi notre classe de collocations (si le nom est variable en nombre ou accepte plusieurs types de déterminant) ou parmi les expressions figées (si le nom est invariable). En revanche, les formes pronominales ne sont pas annotées dans notre projet.

Le processus d'annotation s'est déroulé selon la procédure suivante. Dans un premier temps, trois annotateurs ont annoté les 20 textes selon le guide et nous avons confronté les résultats. Les annotations communes à au moins deux annotateurs ont été retenues et les autres cas ont été discutés par l'équipe pour la création du corpus de référence. Celui-ci ne contenait toutefois que 89 expressions verbales (et il a été utilisé comme test pour se mettre d'accord sur les critères d'annotation). Nous avons complété celui-ci avec un autre corpus de manuels FLE, sélectionnées à partir du corpus 1 et annoté suivant le guide mis à jour après la constitution du corpus de référence. Deux annotateurs ont annoté ce nouveau corpus et ont obtenu un accord inter-annotateur ayant un bon rappel, mais une faible précision (précision : 0,56, rappel :0,97, F-mesure : 0,71). Les expressions idiomatiques et les expressions figées sont souvent correctement annotées par les deux annotateurs. Les divergences entre annotateurs se situent surtout au niveau des collocations. Par exemple, les expressions trop spécifiques à un domaine (*mener une enquête*) ou des expressions qui sont plutôt des combinaisons libres de verbes et de noms (*limiter l'accès*) ont été annoté différemment selon les annotateurs. Au total, 271 EP verbales ont

4. https://typo.uni-konstanz.de/parseme/index.php/2-general/202-parseme-shared-task-on-automatic-identification-of-verbal-mwes-edition-1-1

été identifiées manuellement (41 expressions idiomatiques, 97 collocations, 133 expressions figées). Seulement 81 EP de ce corpus de référence se retrouvent dans notre base, ce qui montre les limites de sa couverture. Notre méthode d'extraction obtient quant à elle un bon rappel (0,77), mais une précision faible (0,43) et une F-mesure de 0,55. Il est assez difficile de comparer ces résultats avec d'autres systèmes et corpus annotés, car les catégories d'expressions et les critères de sélection ne sont pas les mêmes (Ramisch *et al.*, 2018).

Ces résultats sont explicables par le nombre de variantes générées pour les collocations. Cela augmente le risque d'identifier par erreur une suite de mots similaire à une EP, mais qui n'en est pas une (ex. *il a vu le jour en décembre* vs. *il a vu le jour se lever*). Par ailleurs, l'évaluation a été effectuée en ne considérant comme corrects que les cas de reconnaissance exacte des EP présentes dans la base (à l'exception de variations telles qu'un déterminant possessif ou une préposition). Les cas de reconnaissance partielle ont été considérés comme erronés.

Sur la base de cette évaluation, nous pouvons constater que la couverture de PolylexFLE reste limitée : le nombre d'expressions trouvées dans le corpus d'évaluation est très réduit. Par contre, nous avons observé que les annotateurs humains identifient un nombre important d'expressions absentes de notre base, ce qui laisse penser qu'une analyse de nos corpus pédagogiques de plus grande ampleur pourrait se révéler utile pour compléter notre base.

4.3 Comparaison avec des outils d'extraction automatique

Pour évaluer les résultats de notre système, nous comparons les résultats de notre extracteur avec Veyn (Zampieri *et al.*, 2018), outil développé dans le cadre du projet PARSEME-FR. Ce système adopte une approche à base de réseaux de neurones pour l'annotation des expressions polylexicales. Nous avons appliqué Veyn sur le corpus annoté en niveau qui a servi pour construire le corpus de référence (le corpus FLE de 20 textes, qui contient 89 expressions). Sur les 109 expressions annotées par Veyn, seules 8 expressions sont annotés par les deux outils, ce qui est assez surprenant.

Un premier problème qui se pose est la différence entre les catégories d'expressions polylexicales utilisées. Dans le cadre du projet PARSEME-FR, ce sont surtout les expressions idiomatiques et les formes verbales pronominales qui sont annotées. Contrairement à Veyn, qui annote des expressions à verbe support (verbes support avec ou sans nom prédicatif), nous ne détaillons pas ce type d'expressions. En revanche, notre système annote principalement des collocations. PARSEME n'annote pas les collocations, car ils adopent la définition de Sag *et al.* (2001) : les collocations sont des combinaisons fréquentes de mots. Notre définition des collocations se situe plutôt dans la lignée de travaux de Baldwin & Kim (2010) : les collocations ne présentent pas uniquement des combinaisons fréquentes de mots, mais ces mots doivent entretenir des liens syntaxiques et des préférences lexicales fortes, et présenter une variabilité importante (présence des modifieurs, variation des déterminants). Signalons aussi que sur les 109 candidats extraits par Veyn, 43 d'entre-elles sont des formes verbales pronominales, que notre outil ne reconnait pas. En conséquence de ces différentes divergences, on constate peu d'intersections entre les résultats de notre extracteur et ceux de Veyn. Les 8 expressions détectées en commun sont des expressions idiomatiques, la seule catégorie d'expressions véritablement commune aux deux outils.

Un autre problème est celui de la délimitation d'expressions. Ainsi, Veyn connaît parfois des problèmes quand il rencontre une expression polylexicale, car il sélectionne parfois toute la phrase jusqu'à la fin. Ainsi, 15 expressions sur 109 ont été mal délimitées par Veyn.

En résumé, les différences de catégories des EP sont très importantes entre les deux outils, ce qui explique ces résultats divergeants. Il est dès lors compliqué de comparer les résultats des deux systèmes sans une définition commune de ces catégories.

4.4 Des distributions à un niveau CECR unique

Dans la dernière étape de ce projet, qui consistait à attribuer un niveau CECR à chaque EP qui n'en comportait pas encore, nous avons suivi la méthodologie utilisée pour construire FLELex (François *et al.*, 2014). En résumé, une fois les expressions détectées dans les textes du *Corpus-M* décrit à la section 4.1, nous les comptons afin d'obtenir, pour chaque expression, un vecteur de fréquence selon la procédure suivante. Soit une expression E_i de notre collection, celle-ci est associée à un vecteur de fréquences $F_i = (f_{A1}, f_{A2}, f_{B1}, f_{B2}, f_{C1}, f_{C2})$ dans lequel les fréquences sont initialisées à 0. Ensuite, chaque texte du corpus étant associé à l'un des six niveaux du CECR, lorsqu'une expression cible y est trouvée, nous incrémentons de 1 la fréquence f_j du vecteur où j correspond au niveau CECR du texte. À la différence de François *et al.* (2014), nous utilisons uniquement la fréquence relative pour chaque niveau, sans la multiplier par une mesure de dispersion.

La table 2 donne un aperçu des vecteurs obtenus pour quelques expressions. On peut distinguer différents profils fréquenciels pour les expressions de notre liste. Certaines, comme *aller bien* ou *avoir (+nombre) enfant*, sont plutôt typiques des premiers niveaux du CECR et des situations de communication concrètes (ex. se décrire, faire connaissance, etc.) ; d'autres sont plutôt des expressions communément utilisées dans des situations de communication plus professionnelles (ex. *prendre en compte*, *faire partie*) et se retrouvent aux différents niveaux du cadre ; enfin, certaines expressions apparaissent clairement à des stades plus avancés du processus d'apprentissage (ex. *être en droit de*, *avoir tendance*) et correspondent à un usage plus soutenu de la langue. Le principal problème de cette approche est le nombre réduit d'occurrences de nos expressions, vu la petite taille du corpus, qui nuit à la robustesse de l'estimation de ces fréquences.

expression	A1	A2	B1	B2	C1	C2	total
avoir (+nombre) enfant	6	5	0	0	0	0	11
aller bien	71	111	86	2	2	0	272
faire partie	1	2	16	3	7	7	36
prendre en compte	0	1	1	3	2	2	9
être en droit de	0	0	0	0	5	1	6
avoir tendance	0	0	6	2	5	0	13

TABLE 2 – Exemples de vecteurs de fréquence obtenus pour quelques expressions de notre liste.

Une fois les distributions de fréquences estimées, nous les avons transformé en un niveau CECR unique selon les règles suivantes. Si l'EP a été observée au sein d'un seul niveau, nous proposons ce niveau par défaut. C'est le cas de l'expression *faire l'effet d'une bombe* qui n'est observée qu'au niveau C2. Si l'EP est présente dans plusieurs niveaux, nous calculons alors la fréquence relative maximale et nous proposons le niveau où se trouve la fréquence la plus élevée. Ainsi, pour l'expression *faire partie* (*cf.* table 2), le niveau retenu sera B1. Après ce calcul, nous vérifions si l'EP est déjà associée à un niveau CECR dans la liste des expressions obtenue manuellement. Quand ce n'est pas le cas, le niveau obtenu sur le corpus est ajouté à la base. Quand un niveau existe déjà, nous le comparons à celui calculé sur le corpus et conservons systématiquement le niveau obtenu manuellement. Si la

distribution estimée pour une EP est uniforme par niveau, alors la base ne sera pas mise à jour. Pour l'expression *faire partie* nous avons identifié 44,45% d'occurences présentes dans le niveau B1, 19,44 % d'occurences sont présentes dans le niveau C1 et C2 et 8,33 % dans le niveau B2, le reste de 5,55 % est présent dans le niveau A2 et de 2,77 % dans le niveau A1. Dans ce cas, nous avons sélectionné le niveau B1, qui était le plus représenté.

En respectant cette méthodologie, nous avons extrait et annoté automatiquement 580 expressions à partir du *Corpus1* et 506 EP à partir du *Corpus2*. Certaines expressions sont présentes dans les deux corpus et aussi dans la base. Au final, 1098 expressions sont annotées dans la base, à l'aide de l'annotation automatique et manuelle.

5 Évaluation de la qualité des annotations CECR

Au terme de ces deux étapes - manuelle et automatique - nous avons été capable d'attribuer une annotation CECR à 1098 EP parmi les 4295 entrées que comprend la base PolylexFLE. Notre approche comporte toutefois deux faiblesses. D'une part, les annotations de niveau provenant de plusieurs sources tantôt pédagogiques (Beacco, Rey), tantôt des données de corpus, elles courrent le risque d'être partiellement hétérogènes si les critères d'attribution d'une expression à un niveau CECR ne sont pas homogènes. D'autre part, le corpus utilisé pour l'étude étant relativement petit , l'estimation du vecteur de fréquence semble susceptible d'être trop peu robuste. C'est pourquoi, dans cette dernière section, nous avons effectué une expérience préliminaire de la fiabilité pédagogique des annotations CECR de la base PolylexFLE.

Pour réaliser cette évaluation, nous avons constitué un questionnaire composé de 30 expressions sélectionnées selon un échantillonnage aléatoire stratifié (de 4 à 7 EP par niveau du CECR). À l'aide d'un formulaire GoogleForm, il a été demandé à chacun de nos 26 participants volontaires de renseigner son niveau CECR avant de distinguer, parmi les trente expressions, celles qu'il/elle connaît et ne connaît pas. Nous avons ensuite aggrégé les annotations sur le statut connu/inconnu de chaque expression afin de déterminer s'il y avait accord entre le niveau des EP renseignées dans notre base et les connaissances effectives des apprenants.

La distribution des niveaux de compétence parmi nos participants est la suivante : 4 apprenants A1, 11 de niveau A2, 7 participants B1, seulement un B2, 2 C1 et un de niveau C2. Cette distribution n'est pas optimale pour évaluer les EP avancées (B2 à C2), mais il est souvent plus difficile de recruter des participants plus avancés.

Afin de vérifier dans quelle mesure les niveaux CECR de nos EPs sont correctement estimés, nous avons comparé le niveau de chaque participant (N_P) avec celui de l'expression (N_E) et avons agrégé les données de l'expérience de la façon suivante. Pour une EP donnée, nous avons collecté 26 jugements, exprimés par des apprenants dont le niveau varie. Nous les avons répartis au sein de trois classes :

- Inférieure : les apprenants pour lesquels $N_P < N_E$, c'est-à-dire dont le niveau de compétence est inférieur au niveau estimé de l'EP.
- Égale : les apprenants où $N_P = N_E$.
- Supérieure : les apprenants où $N_P > N_E$.

Ensuite, nous avons calculé, au sein de chaune de ces trois classes, le pourcentage d'apprenants qui connaissait l'expression. La situation optimale devrait correspondre à un pourcentage de 0%

pour la classe inférieure (les apprenants ne sont pas encore supposés avoir étudié cette EP) ; à un pourcentage d'environ 50% pour la classe égale (certains apprenants auront déjà rencontré l'EP, tandis que d'autres pas) et 100% pour la classe supérieure (les apprenants d'un niveau de compétence supérieur devraient bien connaître cette expression). Les résultats obtenus sur nos 30 expressions sont toutefois différents de ces pourcentages idéaux (*cf.* Figure 2, quand X vaut 0). Les apprenants de niveau inférieur connaissaient tout de même nos expressions dans $45,6\%$ des cas ; ceux du niveau de compétence égal au niveau de l'EP la connaissaient dans 70% des cas, tandis que ceux de niveau supérieur ne les connaissaient que dans 76% des cas.

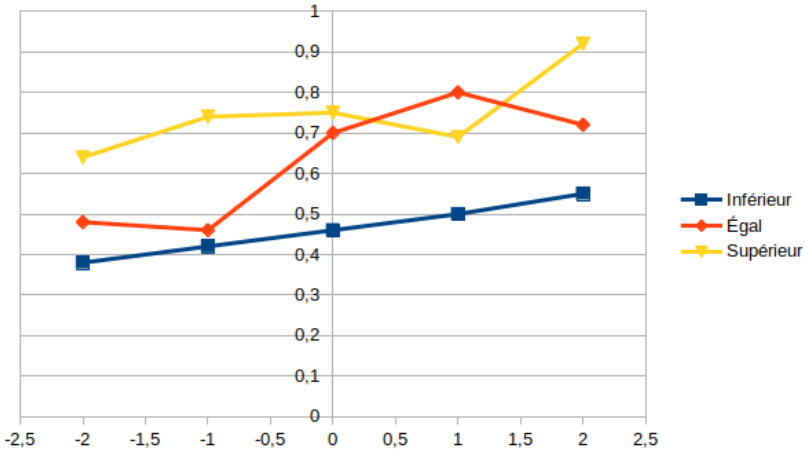

FIGURE 2 – Graphique montrant l'annotation optimale sur la base de notre échantillon

Ces résultats paraissent acceptables, même s'ils sont loins de correspondre à la situation idéale décrite ci-dessus. Afin d'obtenir une comparaison plus réaliste, nous nous sommes demandés quels auraient été ces pourcentages si nous avions classés différemment nos expressions. La Figure 2 montre les pourcentages obtenus dans 5 configurations différentes. Celle de base (X = 0) correspond à l'annotation de PolylexFLE. Dans cette configuration, l'expression *faire partie* est donc classée comme B1. Nous avons ensuite imaginé deux configurations dans lesquelles les expressions auraient été classées un (X = -1), voire deux (X = -2) niveaux en-dessous de la valeur rapportée dans PolylexFLE : ainsi, l'EP *faire partie* y serait respectivement classée comme A2 ou A1. Il y a aussi deux configurations dans lesquelles les EP auraient été classées un (X = 1), voire deux (X = 2) niveaux au-dessous de la valeur rapportée dans PolylexFLE (cad. pour l'EP *faire partie*, comme B2 ou C1). Nous pouvons observer sur la Figure 2 que la configuration qui se rapproche le plus des pourcentages optimaux est celle où la difficulté des EP est d'un niveau inférieur à ceux décrits dans PolylexFLE, en particulier pour la classe "Égale". Il est donc possible que les niveaux estimés dans PolylexFLE soient légèrement sur-évalués. Ce n'est pas totalement surprenant, puisque les expressions tirées de Beacco et ses collègues sont orientés vers la production, alors que notre ressource et notre expérience évaluent les connaissances en réception. Dès lors, le niveau de maîtrise d'une expression en production est logique supérieure à celui de sa maîtrise en réception. Une expérience de plus grande ampleur serait toutefois nécessaire pour confirmer ces résultats préliminaires.

6 Conclusion et perspectives

Pour favoriser l'apprentissage des expressions verbales polylexicales, nous avons construit une base comptant 4295 d'EP verbales, PolylexFLE, dont 1098 se sont vu attribuer un niveau CECR. Ce niveau

est soit obtenu par l'intermédiaire de ressources pédagogiques, soit calculé automatiquement à l'aide d'outils de TAL capables de repérer ces EP verbales dans un corpus lemmatisé et annoté en niveaux CECR à l'aide des patrons morpho-syntaxiques et d'en estimer la distribution. Pour améliorer la qualité de l'extraction, nous utilisons des informations issues d'une analyse syntaxique automatique afin de détecter les paires verbe-objet et vérifier s'il s'agit bien d'expressions polylexicales. Nous avons comparé notre méthode, qui utilise une base lexicale et un système à base de règles avec Veyn, l'outil d'extraction automatique des EP sur le même corpus. Toutefois, cette comparaison s'avère difficile car les catégories choisies dans le projet PARSEME et dans notre projet sont différentes. On constate que les expressions idiomatiques, la seule classe commune aux deux projets, sont reconnus par les deux systèmes. Il y a plus de différences entre collocations et constructions à verbe support. D'autre part, Veyn ne s'adresse pas explicitement aux apprenants et annote les formes verbales pronominales, qui ont un intérêt restreint pour les apprenants d'une langue. Nous n'avons pas pu comparer le corpus annoté dans le cadre du projet PARSEME FR, avec les mêmes textes annotés par notre methode d'identification, car les différences de catégories rendent la tâche très complexe.

La base PolylexFLE présentée dans cet article sera disponible en ligne pour la communauté scientifique via deux sources et sous licence Creative Common. La base PolylexFLE dans son ensemble (4295 entrées) sera rendue disponible sur le site du projet SimpleApprenant [5]. La plateforme qui intègre PolylexFLE sera également, à terme, évaluée par des apprenants de français dans plusieurs universités partenaires selon le protocole suivant : entrainement intensif, écriture des textes avec des expressions aléatoirement générées pour le niveau donné et évaluation du nombre d'erreurs. Par ailleurs, les 1098 entrées pour lesquelles une distribution de fréquence a été calculée se rapprochent très fortement des objectifs des ressources du projet CEFRLex (François *et al.*, 2014, 2016; Tack *et al.*, 2018; Dürlich & François, 2018) et sera rendue disponible sur le site du projet [6].

En ce qui concerne les perspectives ouvertes par cette étude, l'elément qui nous paraît le plus important est celui qui dérive de la méthodologie proposée. Celle-ci pourrait en effet être facilement adaptée à d'autres données, en particulier des données produites par des apprenants du FLE. Il serait alors possible de comparer les distributions de fréquence des EP du français en contexte de réception (ex. lecture) et de production. Une autre piste qui mériterait d'être approfondie est de modifier la fonction de discrétisation qui a servi à transformer les distributions de fréquence en un niveau unique. La procédure utilisée, à savoir la comparaison des fréquences par niveau, pourra être améliorée en normalisant les fréquences par une mesure de dispersion, suivant la méthodologie appliquée pour FLELex (François *et al.*, 2014).

Remerciements

Le projet SimpleApprenant a été financé par le programme IdEx de l'Université de Strasbourg pour la période de juin 2017 à février 2019. Le site Web du projet SimpleApprenant est hébérgé par la TGIR Huma-Num. Une première version du script d'extraction a été dévelopé par Colm Stapleton. Nous remercions nos partenaires de l'Université d'Opole (Pologne) (Mme Magda Danko et M. Fabrice Marsac) et de l'Université de Chypre (Mme Fabienne Baider et Mme Marina Christofi) pour leur aide précieux pour la mise en place des évaluations pédagogiques.

5. https ://simpleapprenant.huma-num.fr/SimplifyYourFrench/accueil
6. http ://cental.uclouvain.be/cefrlex/

Références

BAHNS J. & ELDAW M. (1993). Should We Teach EFL Students Collocations ? *System*, **21**(1), 101–14.

BALDWIN T. & KIM S. N. (2010). Multiword Expressions. In *Handbook of natural language processing*, p. 267–292. Boca Raton, FL : CRC Press, Taylor and Francis Group, 2 edition.

BEACCO J.-C., BOUQUET S. & PORQUIER R. (2004). *Niveau B2 pour le français : un référentiel : utilisateur-apprenant indépendant*. Mayenne : Didier.

BEACCO J.-C., LEPAGE S., PORQUIER R. & RIBA P. (2008). *Niveau A1 pour le français : utilisateur-apprenant élémentaire*. Mayenne : Didier.

CAVALLA C., LOISEAU M., DIWERSY S., LASCOMBE V. & SOCHA J. (2013). EmoProf. In *Journées Lig-Lidilem*, Eybens (Grenoble), France.

COAVOUX M. & CRABBÉ B. (2017). Incremental discontinuous phrase structure parsing with the gap transition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 1259–1270, Valencia, Spain : Association for Computational Linguistics.

CONSEIL DE L'EUROPE (2001). *Cadre européen commun de référence pour les langues : apprendre, enseigner, évaluer*. Paris : Hatier.

CONSTANT M., ERYIĞIT G., MONTI J., VAN DER PLAS L., RAMISCH C., ROSNER M. & TODIRASCU A. (2017). Multiword Expression Processing : A Survey. *Computational Linguistics*, **43**(4), 837–892.

DIWERSY S., GOOSSENS V., GRUTSCHUS A., KERN B., KRAIF O., MELNIKOVA E. & NOVAKOVA I. (2014). Traitement des lexies d'émotion dans les corpus et les applications d'emobase. *Corpus*, **13**, 269–293.

DÜRLICH L. & FRANÇOIS T. (2018). EFLLex : A Graded Lexical Resource for Learners of English as a Foreign Language. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, p. 873–879.

FRANÇOIS T., GALA N., WATRIN P. & FAIRON C. (2014). FLELex : a graded lexical resource for French foreign learners. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, p. 3766–3773.

FRANÇOIS T., VOLODINA E., ILDIKÓ P. & TACK A. (2016). SVALex : a CEFR-graded lexical resource for Swedish foreign and second language learners. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, p. 213–219.

FRANÇOIS T. (2014). An analysis of a french as a foreign language corpus for readability assessment. In *Proceedings of the 3rd workshop on NLP for Computer-assisted Language Learning, NEALT Proceedings Series Vol. 22, Linköping Electronic Conference Proceedings 107*, p. 13–32.

GROSS M. (1993). Les phrases figées en français. *L'information grammaticale*, **59**, 36–41.

GROSS M. (1994). Constructing Lexicon-grammars. In R. ATKINS & A. ZAMPOLLI, Eds., *Computational Approaches to the Lexicon*, p. 213–263, Oxford : Oxford Univ. Press.

HAMEL M.-J. & MILICEVIC J. (2007). Analyse d'erreurs lexicales d'apprenants du fls : démarche empirique pour l'élaboration d'un dictionnaire d'apprentissage. *Canadian Journal of Applied Linguistics*, **10**(1), 25–45.

HAMEL M.-J., SLAVKOV N., INKPEN D. & XIAO D. (2016). Myannotator : A tool for technology-mediated written corrective feedback. *Traitement Automatique des Langues*, **57**(3), 119–142.

HATHOUT N., SAJOUS F. & CALDERONE B. (2014). GLÀFF, a Large Versatile French Lexicon. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, p. 1007–1012, Reykjavik, Iceland.

KREMMEL B., BRUNFAUT T. & ALDERSON J. C. (2017). Exploring the role of phraseological knowledge in foreign language reading. *Applied Linguistics*, **38**(6), 848–870.

LAPORTE E., RANCHHOD E. & YANNACOPOULOU A. (2008). Syntactic variation of support verb constructions. *Lingvisticae Investigationes*, **31**(2), 173–185. DOI : 10.1075/li.31.2.04lap.

MADNANI N., BURSTEIN J., SABATINI J., BIGGERS K. & ANDREYEV S. (2016). Language MuseTM : Automated Linguistic Activity Generation for English Language Learners. In R. ATKINS & A. ZAMPOLLI, Eds., *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, p. 213–263, Berlin : ACL.

PAWLEY A. & SYDER F. (1983). Two puzzles for linguistic theory : nativelike selection and nativelike fluency. In J. RICHARDS & R. SCHMITT, Eds., *Language and Communication*, p. 191–225. London : Longman.

RAMISCH C., CORDEIRO S., SAVARY A., VINCZE V., MITITELU V., BHATIA A., BULJAN M., CANDITO M., GANTAR P., GIOULI V. *et al.* (2018). Edition 1.1 of the parseme shared task on automatic identification of verbal multiword expressions. In *The Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, p. 222–240.

REY I. G. (2007). *La didactique du français idiomatique*. Editions Modulaires Européennes InterCommunication.

SAG I., BALDWIN T., BOND F., COPESTAKE A. & FLICKINGER D. (2001). Multiword expressions : A pain in the neck for nlp. In *In Proc. of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002*, p. 1–15.

TACK A., FRANÇOIS T., DESMET P. & FAIRON C. (2018). NT2Lex : A CEFR-Graded Lexical Resource for Dutch as a Foreign Language Linked to Open Dutch WordNet. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications (NAACL 2018)*.

TODIRASCU A. & CARGILL M. (2019). SimpleApprenant : a Platform to Assist French L2 Language Learners to Improve Writing Skills. In *Proceedings of the 27th EUROCALL conference (to appear)*, Louvain-La-Neuve, Belgique.

TUTIN A., ESPERANÇA-RODIER E., IBORRA M. & REVERDY J. (2015). Annotation of multiword expressions in French. In C.-P. GLORIA, Ed., *European Society of Phraseology Conference (EUROPHRAS 2015)*, Computerized and Corpus-based Approaches to Phraseology : Monolingual and Multilingual Perspectives, p. 60–67, Malaga, Spain.

VERLINDE S., BINON J. & SELVA T. (2006). The base lexicale du français (blf) : A multifunctional online database for learners of french. In C. O. ELISA CORINO, CARLA MARELLO, Ed., *Proceedings of the 12th EURALEX International Congress*, p. 471–481, Torino, Italy : Edizioni dell'Orso.

ZAMPIERI N., SCHOLIVET M., RAMISCH C. & FAVRE B. (2018). Veyn at parseme shared task 2018 : Recurrent neural networks for vmwe identification. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, p. 290–296, Santa Fe, New Mexico, USA : Association for Computational Linguistics.