Extraction et fouille de données textuelles : application à la détection de la dépression, de l'anorexie et de l'agressivité dans les réseaux sociaux

Faneva Ramiandrisoa

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Cotutelle internationale : Université d'Antananarivo

Présentée et soutenue par

Iarivony RAMIANDRISOA

Le 14 décembre 2020

Extraction et fouille de données textuelles: application à la détection de la dépression, de l'anorexie et de l'agressivité dans les réseaux sociaux.

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Josiane MOTHE et Michel Martin RAJOELINA

Jury

Mme Béatrice DAILLE, Rapporteure
M. Vincent CLAVEAU, Rapporteur
M. Eric SANJUAN, Examinateur
M. Max CHEVALIER, Examinateur
M. Adrian-Gabriel CHIFU, Examinateur
Mme Josiane MOTHE, Directrice de thèse
M. Michel Martin RAJOELINA, Co-directeur de thèse

Remerciements

Comme préambule à ce manuscrit, je souhaiterais adresser mes remerciements les plus sincères aux personnes qui m'ont, de près ou de loin, apporté leur aide et qui ont contribué à l'élaboration de ce manuscrit ainsi qu'à l'aboutissement de cette thèse.

Je voudrais tout d'abord remercier les membres du jury. Merci à mes rapporteurs, Mme Béatrice DAILLE et M. Vincent CLAVEAU, pour leur travail de relecture. Merci à mes examinateurs, MM. Max CHEVALIER, Eric SANJUAN et Adrian-Gabriel CHIFU pour avoir accepté d'évaluer mon travail de thèse. Merci à tous de vous être intéressés à mon travail, d'avoir pris le temps de lire mon manuscrit, et d'avoir participé à ma soutenance.

Ce qui m'amène aux autres membres du jury, mes directeurs de thèse, qui se sont toujours montrés à l'écoute et très disponible tout au long de la thèse. Merci à la Pr. Josiane MOTHE, d'abord pour l'encadrement à distance de mon mémoire de Master (à Madagascar) puis pour ces années de thèse passées sous sa direction. Je la remercie pour son suivi permanent, ses commentaires et ses conseils qui m'ont été précieux tout au long de ce travail. Merci au Pr. Michel Martin RAJOELINA, d'avoir accepté de diriger ma thèse malgré le fait que nous travaillons dans des domaines de recherches très différents. Je le remercie pour ses précieuses aides, sa disponibilité et ses conseils tout au long de la thèse. Outre les soutiens scientifiques de mes directeurs de thèse, leurs conseils m'ont permis d'avancer sereinement dans mes travaux. Je tiens à leur exprimer toute ma gratitude et ma reconnaissance.

Je tiens à remercier l'équipe SIG pour l'accueil que j'ai reçu. Merci aux permanents, aux doctorants et Postdocs de cette équipe formidable. Merci à Md Zia Ullah, Nathalie Neptune, Clément Lejeune, Nabil El Malki et les autres que je n'ai pas pu citer. Nous avons passé de très bons moments ensemble : collaborations, pauses café, etc. Je remercie aussi tout le personnel du laboratoire IRIT pour leur gentillesse et les services qu'ils m'ont apportés.

J'adresse un remerciement au Pr. Norbert Fuhr qui m'a accueilli dans son équipe lors de mon séjour en Allemagne. Je le remercie pour ses conseils que j'ai pu appliquer dans mon travail. Merci aussi à toute son équipe, à ses doctorants et à son Postdoc qui m'ont très bien accueilli et m'ont aidé lors de ce séjour en Allemagne.

Je tiens à exprimer ma gratitude aux Malgaches que j'ai rencontrés en France, surtout sur Toulouse, pour leur aide et soutien durant mon séjour à Toulouse. Grâce à vous, je ne me sentais jamais loin de ma famille.

Je tiens aussi à remercier M. Tahiry ANDRIAMAROZAKANIAINA, responsable du

parcours Master de la MISA, et M. Andry RASOANAIVO, directeur de la MISA, qui ont accepté de répondre à toutes mes questions avec gentillesse et m'ont permis d'accéder à des salles de la MISA pour effectuer mes recherches. Merci à M. Patrick RABARISON d'avoir pris la peine de lire mon manuscrit et de suggérer des améliorations. Merci aussi aux personnes qui ont partagé leur bureau avec moi durant mes recherches à Madagascar. Je remercie aussi les amis, le corps enseignant et les étudiants de la MISA.

Enfin, je tiens à remercier les membres de ma famille et mes proches pour leur soutien au cours de ces longues années d'études. En particulier à ma mère pour son amour inconditionnel et pour son soutien en toutes circonstances.

Merci à toutes et à tous.

Résumé

Notre recherche porte essentiellement sur des tâches ayant une finalité applicative : détection de la dépression et de l'anorexie d'une part et détection de l'agressivité d'autre part ; cela à partir de messages postés par des utilisateurs de plates-formes de type réseaux sociaux. Nous avons également proposé une méthode non supervisée d'extraction de termes-clés.

Notre première contribution porte sur l'extraction automatique de termes-clés dans des documents scientifiques ou articles de presse. Plus précisément, nous améliorons une méthode non supervisée à base de graphes. Nous avons évalué notre approche sur onze collections de données dont cinq contenant des documents longs, quatre contenants des documents courts et enfin deux contenant des documents de type article de presse. Nous avons montré que notre proposition permet d'améliorer les résultats dans certains contextes.

La deuxième contribution de cette thèse est une solution pour la détection au plus tôt de la dépression et de l'anorexie. Nous avons proposé des modèles utilisant des classifieurs, s'appuyant sur la régression logistique ou les forêts d'arbres de décision, basés sur (a) des caractéristiques et (b) le plongement de phrases. Nous avons évalué nos modèles sur les collections de données de la tâche eRisk. Nous avons observé que les modèles basés sur les caractéristiques sont très performants lorsque la mesure de précision est considérée, soit pour la détection de la dépression, soit pour la détection de l'anorexie. Le modèle utilisant le plongement de phrases, quant à lui, est plus performant lorsque l'on mesure la détection au plus tôt ($ERDE_{50}$) et le rappel. Nous avons aussi obtenu de bons résultats par rapport à l'état de l'art : meilleurs résultats sur la précision et $ERDE_{50}$ pour la détection de la dépression, et sur la précision et le rappel pour la détection de l'anorexie.

Notre dernière contribution concerne la détection de l'agression dans les messages postés par des utilisateurs sur les réseaux sociaux. Nous avons réutilisé les mêmes modèles que ceux utilisés pour la détection de la dépression ou de l'anorexie. À cela, nous avons ajouté d'autres modèles basés sur l'apprentissage profond. Nous avons évalué nos modèles sur les collections de données de la tâche internationale TRAC. Nous avons observé que nos modèles, utilisant l'apprentissage profond, fournissent de meilleurs résultats que nos modèles utilisant des classifieurs classiques. Nos résultats dans cette partie de la thèse sont comparables à l'état de l'art du domaine. Nous avons toutefois obtenu le meilleur résultat sur une des collections de données.

Abstract

Our research mainly focuses on tasks with an application purpose : depression and anorexia detection on the one hand and aggression detection on the other ; this from messages posted by users on a social media platform. We have also proposed an unsupervised method of keyphrases extraction. These three pieces of work were initiated at different times during this thesis work.

Our first contribution concerns the automatic keyphrases extraction from scientific documents or news articles. More precisely, we improve an unsupervised graph-based method to solve the weaknesses of graph-based methods by combining existing solutions. We evaluated our approach on eleven data collections including five containing long documents, four containing short documents and finally two containing news articles. We have shown that our proposal improves the results in certain contexts.

The second contribution of this thesis is to provide a solution for early depression and anorexia detection. We proposed models that use classical classifiers, namely logistic regression and random forest, based on : (a) features and (b) sentence embedding. We evaluated our models on the eRisk data collections. We have observed that feature-based models perform very well on precision-oriented measures both for depression or anorexia detection. The model based on sentence embedding is more efficient on $ERDE_{50}$ and recall-oriented measures. We also obtained better results compared to the state-of-the-art on precision and $ERDE_{50}$ for depression detection, and on precision and recall for anorexia detection.

Our last contribution is to provide an approach for aggression detection in messages posted by users on social networks. We reused the same models used for depression or anorexia detection to create models. We added other models based on deep learning approach. We evaluated our models on the data collections of TRAC shared task. We observed that our models using deep learning provide better results than our models using classical classifiers. Our results in this part of the thesis are in the middle (fifth or ninth results) compared to the competitors. We still got the best result on one of the data collections.

Publications

La recherche rapportée dans cette thèse a abouti aux publications suivantes :

Articles de revues nationales :

1. Farah Benamara, Véronique Moriceau, Josiane Mothe et Faneva Ramiandrisoa. Aide à la détection automatique des utilisateurs dépressifs dans les médias sociaux. Document Numérique, vol. 22, no. 3, pages 49–74, 2019.

Conférences et workshops internationaux :

1. Faneva Ramiandrisoa et Josiane Mothe. IRIT at TRAC 2020. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 49–54, 2020.
2. Faneva Ramiandrisoa et Josiane Mothe. Aggression Identification in Social Media : a Transfer Learning Based Approach. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 26–31, 2020.
3. Faneva Ramiandrisoa et Josiane Mothe. Early Detection of Depression and Anorexia from Social Media : A Machine Learning Approach. In Proceedings of the Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020, 2020.
4. Faneva Ramiandrisoa. Aggression Identification in Posts – two machine learning approaches. In Workshop on Machine Learning for Trend and Weak Signal Detection in Social Networks and Social Media., Toulouse, France, 2020. CEUR-WS.org.
5. Razan Masood, Faneva Ramiandrisoa et Ahmet Aker. UDE at eRisk 2019 : Early Risk Prediction on the Internet. In Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019.
6. Josiane Mothe, Faneva Ramiandrisoa et Michael Rasolomanana. Automatic keyphrase extraction using graph-based methods. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018, pages 728–730, 2018.
7. Faneva Ramiandrisoa et Josiane Mothe. IRIT at TRAC 2018. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018, pages 19–27, 2018.

8. Faneva Ramiandrisoa, Josiane Mothe, Farah Benamara et VéroniqueMoriceau. IRIT at e-Risk 2018. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018.
9. Idriss Abdou Malam, Mohamed Arziki, Mohammed Nezar Bellazrak, Farah Benamara, Assafa El Kaidi, Bouchra Es-Saghir, Zhaolong He, Mouad Housni, Véronique Moriceau, Josiane Mothe et Faneva Ramiandrisoa. IRIT at e-Risk. In Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September11-14, 2017.

Conférences et workshops nationaux :

1. Farah Benamara, Véronique Moriceau, Josiane Mothe, Faneva Ramiandrisoa et Zhaolong He. Automatic Detection of Depressive Users in Social Media. In COnférence en Recherche d'Informations et Applications - CORIA 2018, 15th French Information Retrieval Conference, Rennes, France, May 16-18, 2018. Proceedings, 2018.
2. Faneva Ramiandrisoa et Josiane Mothe. Profil utilisateur dans les réseaux sociaux : Etat de l'art. In COnférence en Recherche d'Informations et Applications - CORIA 2017, 14th French Information Retrieval Conference, Marseille, France, March 29-31, 2017. Proceedings, pages 395–404, 2017.
3. Josiane Mothe et Faneva Ramiandrisoa. Extraction automatique de termes-clés : Comparaison des méthodes non supervisées de la littérature. In CORIA 2016 - Conférence en Recherche d'Informations et Applications - 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, pages 315–324, 2016.

Séminaire national :

1. Josiane Mothe, Michel Rajoelina, Faneva Ramiandrisoa et Hary Razakasoa. Intégration des plongements de mots dans les méthodes, supervisées et non supervisées, d'extraction automatique de mots clés. VSST 2018.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

Tf-Idf Term Frequency-Inverse Document Frequency

TPR Topical PageRank

STPR Single Topical PageRank

LDA Latent Dirichilet Allocation

SVM Séparateurs à Vaste Marge

TAL Traitement Automatique des Langues

RI Recherche d'Information

CBOW Countinious Bag of Words

CSLM Continuous Space Language Models

LSA Latent Semantic Analysis

CRF Conditional Random Fields

TANN Topic-based Adversarial Neural Network

Bi-LSTM Bidirectional Long Short-TermMemory

LSTM Long Short-TermMemory

ToLDA Task-oriented Latent Dirichlet Allocation model

RAKE Rapid Automatic Keyword Extraction

MMR Maximal Marginal Relevance

BERT Bidirectional Encoder Representation from Transformer

CNN Convolutional Neural Network (réseau de neurones convolutionnels)

RNN Recurrent Neural Networks (réseaux de neurones récurrents)

FOG Gunning Fog Index

FRE Flesch Reading Ease

LWR Linsear Write Formula

DCR New Dale-Chall Readability

TVT Variation Temporelle des Termes ou Temporal Variation of Terms

FTVT Variation Temporelle Flexible des Termes ou Flexible Temporal Variation of Terms

SIC Classification Séquentielle Incrémentale ou Sequential Incremental Classification

ERDE Early Risk Detection Error

Pals Projected Attention Layers ou Couches d'Attentions Projetées

Chapitre 1

Introduction générale

Ces dernières années, Internet est devenu l'un des principaux outils de communication dans le monde. Le nombre d'utilisateurs sur Internet a augmenté de 83 % de 2014 à 2019 [Samghabadi 2020b]. En avril 2020, près de 4,57 milliards de personnes sont des utilisateurs actifs sur Internet, soit 59 % de la population mondiale. Actuellement, il existe plusieurs plates-formes de médias sociaux, tels que Facebook, Twitter, etc., où les gens peuvent facilement partager des informations et interagir les uns avec les autres dans un espace pratiquement illimité [Samghabadi 2020a].

Avec l'utilisation accrue d'Internet, le volume de données ne cesse d'augmenter et les données sont aujourd'hui au cœur de toute activité. L'exploitation de ces données est devenue un enjeu majeur dans le monde professionnel, dans le monde de la recherche, etc. Ces données sont l'opportunité de nombreux thèmes de recherche comme la détection des risques de suicide en ligne, la détection de fausses informations dans les réseaux sociaux, les applications dans le domaine médical, etc.

Nos travaux dans cette thèse sont volontairement tournés vers des finalités applicatives : détection de la dépression et de l'anorexie d'une part et détection de l'agressivité d'autre part ; cela à partir de messages postés par des utilisateurs de plates-formes de type réseaux sociaux. Nous avons également proposé une amélioration d'une méthode non supervisée d'extraction de termes-clés. Ces trois groupes de travaux ont été initiés dans des moments différents du travail de thèse et ont chacun fait l'objet de publications. Nous avons fait le choix de présenter ces trois groupes de travaux dans des chapitres avec des états de l'art séparés, compte tenu de leur relative indépendance. Lorsque c'est possible, nous avons toutefois montré les liens ou les ponts possibles mais non réalisés faute de temps.

La première contribution de cette thèse porte sur l'extraction automatique de termes-clés dans des documents scientifiques ou articles de presse, plus précisément, nous améliorons une méthode non supervisée à base de graphes qui est celle de Boudin *et al.* [Boudin 2013a]. Nous nous sommes intéressés aux méthodes à base de graphes du fait de leur bonne performance, leur simplicité, la facilité de leur implémentation. Elles sont aussi les plus courantes et sont suffisamment diversifiées. Nous avons constaté que ces méthodes à base de graphes présentent des faiblesses : (1) la construction du graphe

de mots est basée sur la co-occurrence qui ne capture pas très bien la relation sémantique entre deux mots ; (2) l'attribution de scores aux termes-clés candidats est réalisée en fonction de leur longueur, ce qui favorise souvent les candidats les plus longs ou les plus courts ; (3) le calcul des scores des termes-clés candidats dépend des scores des mots qui les composent, ce qui favorise la présence de chevauchements des termes-clés, c'est-à-dire que certains candidats sont des sous-chaînes d'autres candidats ; et enfin (4) la construction du graphe de mots étant basée sur la co-occurrence, cela entraîne la perte d'informations telles que la fréquence ou la position des termes-clés candidats dans le document alors que ces informations peuvent être très importantes dans le choix des termes-clés. Nous résolvons ces quatre problèmes en combinant différentes solutions proposées dans la littérature. Pour le problème (1), nous proposons d'utiliser le plongement de mots lors de la construction du graphe de mots. Puis nous adoptons une modification de la moyenne harmonique [Yeom 2019] pour calculer les scores des termes-clés candidats afin de résoudre le problème (2). Enfin pour résoudre les problèmes (3) et (4), nous recalculons les scores des termes-clés candidats en utilisant la méthode modifiée C-value proposée par Yeom *et al.* [Yeom 2019].

La deuxième contribution de cette thèse est une solution pour la détection au plus tôt de la dépression à partir des publications des utilisateurs sur les réseaux sociaux. L'objectif dans cette partie de recherche est de détecter si un utilisateur est dépressif, en utilisant le moins de publications possible de cet utilisateur, ses publications étant ordonnées par ordre chronologique. Pour atteindre cet objectif, nous proposons des modèles utilisant des classifieurs, s'appuyant sur la régression logistique ou les forêts d'arbres de décision, basés sur (a) des caractéristiques et (b) le plongement de phrases. Nos modèles ont été évalués sur les collections de données des deux éditions de la tâche eRisk (2017 et 2018). L'édition 2018 de la tâche eRisk propose aussi de résoudre le problème de la détection au plus tôt de l'anorexie. Ainsi, nous avons aussi utilisé nos modèles pour résoudre ce problème afin de voir leur portabilité sur des problèmes autre que la dépression.

La dernière contribution de cette thèse concerne la détection de l'agressivité dans les messages postés par des utilisateurs sur les réseaux sociaux. L'objectif dans cette partie de la thèse est de catégoriser une publication postée par un utilisateur selon qu'elle contient de l'agressivité ou non. Pour répondre à cet objectif, nous avons réutilisé les mêmes modèles que ceux utilisés pour la détection de la dépression ou de l'anorexie. À cela, nous avons ajouté d'autres modèles basés sur l'apprentissage profond (CNN, LSTM et BERT).

Cette thèse est divisée en cinq chapitres répartis comme suit :

Le chapitre 1 est cette introduction qui présente nos contributions dans ce travail

de recherche.

Chacun des chapitres suivants correspond à une contribution de nos travaux. Dans ces chapitres, après une introduction du problème, nous présentons l'état de l'art, nos propositions et les évaluations sur des collections de référence internationale.

Le chapitre 2 présente les améliorations que nous avons apportées à une méthode à base de graphes d'extraction de termes-clés.

Le chapitre 3 propose une solution pour la détection au plus tôt de la dépression et de l'anorexie à partir de messages postés par des utilisateurs sur les plates-formes de type réseaux sociaux.

Le chapitre 4 présente notre approche pour la détection de l'agressivité dans les messages postés par des utilisateurs sur les réseaux sociaux.

Le chapitre 5 conclut ce travail de thèse, discute de nos contributions et présente quelques perspectives pour de futurs travaux.

Chapitre 2

Extraction automatique des termes-clés

Sommaire

Résumé.

Les termes-clés jouent un rôle important dans les systèmes automatiques, par exemple pour la recherche d'informations, les études scientométriques, les revues de littérature ou la classification de documents. Actuellement, le volume de documents numériques est tel que l'attribution manuelle des termes-clés est très coûteuse en temps. L'extraction automatique de termes-clés est donc devenue importante. Dans la littérature, plusieurs méthodes, certaines supervisées et d'autres non supervisées, ont été proposées afin d'extraire automatiquement les termes-clés d'un texte. Dans ce travail, nous nous intéressons aux méthodes non supervisées d'extraction automatique de termes-clés à base de graphes. Cependant, les méthodes de la littérature à base de graphes présentent des faiblesses. Nous essayons de résoudre ces faiblesses en apportant une modification à la méthode à base de graphes de Boudin *et al.* [Boudin 2013a]. Nous avons choisi cette méthode parce qu'elle utilise des algorithmes simples pour l'ordonnancement des nœuds du graphe et offre des performances équivalentes aux autres méthodes à base de graphes. Pour l'évaluation de notre approche, nous avons utilisé onze collections de données dont cinq contenant des documents longs, quatre contenant des documents courts et enfin deux contenant des documents de type article de presse. Nous avons montré que notre méthode à base de graphes améliore significativement la méthode de Boudin *et al.* [Boudin 2013a] en considérant la f1-mesure (F) sur tous les corpus de documents longs, seulement sur un corpus de documents courts et aucune amélioration sur le corpus documents de type article de presse. Nous avons aussi comparé notre approche à douze autres méthodes non supervisées de l'état de l'art et avons obtenu comme résultats : résultats équivalents aux meilleures méthodes sur deux des cinq corpus de documents longs et sur un corpus de documents courts, deuxième meilleur résultat sur deux autres corpus de documents longs, deux corpus de documents courts et un corpus de documents de type article de presse, et enfin troisième meilleur résultat sur le dernier corpus de documents longs et le corpus de documents courts.

2.1 Introduction

L'extraction des termes-clés (appelés aussi mots-clés) est la sélection des unités textuelles importantes dans un document [Turney 2000] et caractérisant son contenu principal.

Des termes-clés sont souvent associés aux publications afin d'aider les utilisateurs à avoir une vue d'ensemble et rapide du document. Ces termes peuvent également être utilisés comme une entrée de recherche dans les moteurs de recherche, dans le traitement du langage naturel et l'exploration de texte. Ils peuvent aussi être utilisés pour l'indexation et la recherche de documents scientifiques [Boudin 2020]. L'attribution des termes-clés à un document peut être manuelle (choisis par les auteurs par exemple) ou automatique (comme dans les moteurs de recherche).

Si nous considérons l'exemple des articles scientifiques, il existe trois types de termes associés aux articles : (a) les termes-clés qui sont généralement demandés aux auteurs, (b) les termes-clés choisis par les documentalistes ou par les auteurs à partir d'un thésaurus ou d'une ressource ontologique, (c) les termes-clés extraits automatiquement du contenu du document qui sont soit des mots simples, soit des groupes de mots.

Ces trois types de termes-clés jouent un rôle important dans les systèmes automatiques, par exemple pour la recherche d'informations, les études scientométriques, les revues de littérature ou la classification de documents. Tout en ayant des objectifs communs, les trois types diffèrent fortement. Les termes-clés des auteurs décrivent bien le contenu même du document car les auteurs sont des spécialistes du sujet sur lequel ils écrivent ; mais le choix des termes est très subjectif et différentes variantes peuvent être utilisées pour différents textes sur le même sujet. Les termes-clés extraits de ressources ontologiques ou de thésaurus n'ont pas cet inconvénient car ils sont choisis dans une liste limitée de termes. En revanche, les ressources ontologiques sont difficiles à mettre à jour et leur contenu peut ne pas refléter l'état actuel des connaissances d'un domaine ou ne pas inclure de termes spécifiques qui seraient utiles pour décrire avec précision le contenu du document. Enfin, les termes-clés extraits automatiquement correspondent davantage à des termes d'indexation et sont quelquefois non compréhensibles par les humains du fait des prétraitements utilisés (par exemple, l'utilisation de la racinisation des mots). D'autres techniques au contraire préservent l'intelligibilité des résultats.

Nous nous intéressons à ce dernier type de termes-clés compréhensibles par les humains mais également extraits de façon automatique. En effet, le volume de documents numériques est tel que l'attribution manuelle des termes-clés est très coûteuse en temps. L'extraction automatique de termes-clés ou de mots-clés a pour objectif d'ex-

traire automatiquement un nombre limité d'unités textuelles à partir du document, sans utiliser de ressource ontologique. De manière générale, l'étape d'extraction automatique de termes-clés peut être divisée en plusieurs étapes comme suit : une phase de prétraitement, la sélection des termes-clés candidats, l'ordonnancement ou la classification des candidats et la sélection des termes-clés parmi les candidats.

L'étape de prétraitement est souvent constituée de traitements linguistiques comme le nettoyage du document en supprimant les mots vides (article, pronoms, etc.). Dans les méthodes que nous allons voir et utiliser, la phase d'étiquetage des mots est importante. Il s'agit de reconnaître la classe morpho-syntaxique d'un mot. Cet étiquetage est souvent utilisé lors de l'extraction des termes-clés candidats, qui sont des mots ou expressions (groupe de mots) susceptibles de représenter le document, c'est-à-dire de devenir des termes-clés. À partir de ces termes-clés candidats, des algorithmes d'ordonnancement ou de classification sont utilisés pour décider si le candidat sera considéré comme terme-clé ou non.

La figure 2.1 présente les étapes de la majorité des méthodes automatiques d'extraction de termes-clés.

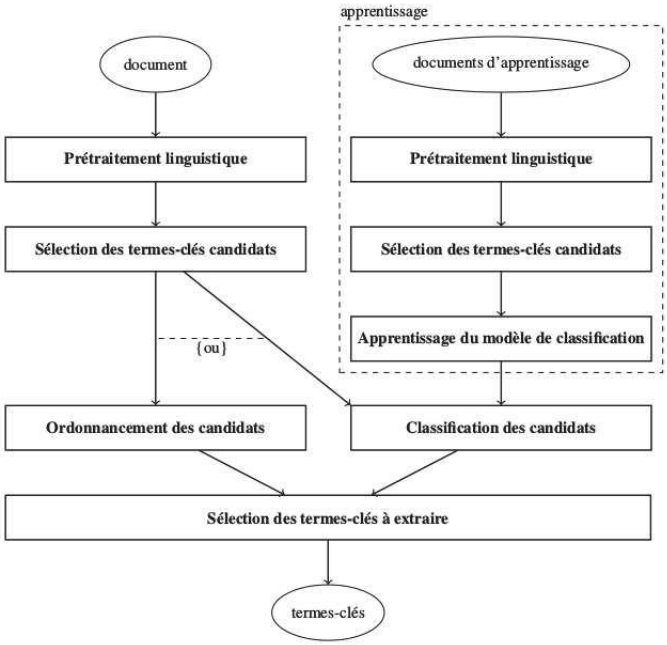


FIGURE 2.1 – Les principales étapes d'extraction automatique de termes-clés [Bougouin 2013b]

Dans la littérature, plusieurs méthodes, certaines supervisées [Witten 1999, Sarkar 2010, Zhang 2006] et d'autres non supervisées [Boudin 2018, Boudin 2013a,

Mihalcea 2004], ont été proposées afin d'extraire automatiquement les termes-clés. Ces méthodes sont formulées soit comme un problème d'ordonnancement (il s'agit alors d'ordonner les termes-clés candidats selon leur importance par rapport au document), soit comme un problème de classification (il s'agit alors de classifier les termes-clés candidats comme termes-clés ou non termes-clés).

L'ordonnancement est souvent réalisé par une méthode non supervisée alors que la classification est réalisée par une méthode supervisée (les méthodes supervisées peuvent aussi réaliser des ordonnancements). L'avantage des méthodes non supervisées par rapport aux méthodes supervisées est qu'elles n'ont pas besoin d'un ensemble d'apprentissage (termes associés manuellement) ; par conséquent, elles sont moins sensibles aux changements de thèmes et donc plus adaptables. Les méthodes supervisées quant à elles sont performantes, mais dépendent fortement de la qualité et de la disponibilité des corpus d'entraînement qui doivent être souvent du même domaine disciplinaire et de la même langue que les documents à traiter [Bougouin 2013b].

Dans ce travail, nous nous intéressons aux méthodes non supervisées d'extraction automatiques de termes-clés. De nombreuses méthodes non supervisées sont à base de graphes du fait de leur bonne performance malgré leur simplicité et la facilité de leur implémentation ; elles sont aussi les plus courantes et sont suffisamment diversifiées.

Ces méthodes représentent le document en graphe de mots pour ordonner les termes-clés candidats extraits du document afin d'en sélectionner les termes-clés. Cependant, ces méthodes présentent des faiblesses : (1) la construction du graphe de mots est basée sur la co-occurrence qui ne capture pas très bien la relation sémantique entre deux mots ; (2) l'attribution de scores aux termes-clés candidats est réalisée en fonction de leur longueur, ce qui favorise souvent les candidats les plus longs ou les plus courts ; (3) le calcul des scores des termes-clés candidats dépend des scores des mots qui les composent, ce qui favorise la présence de chevauchements des termes-clés, c'est-à-dire que certains candidats sont des sous-chaînes d'autre candidats ; et enfin (4) la construction du graphe de mots étant basée sur la co-occurrence, cela entraîne la perte d'informations telles que la fréquence ou la position des termes-clés candidats dans le document alors que ces informations peuvent être très importantes dans le choix des termes-clés.

Des chercheurs ont proposé des solutions pour résoudre ces problèmes mais séparément. En effet, Rui Wang *et al.* [Wang 2015] résolvent seulement le problème (1) en utilisant le plongement de mots lors de la construction du graphe de mots. Yeom *et al.* [Yeom 2019], quant à eux, résolvent le problème (2) en adoptant une moyenne harmonique modifiée pour calculer les scores des termes-clés candidats au lieu de la somme ou la moyenne qu'utilisent les autres méthodes de l'état de l'art. Yeom *et al.*

[Yeom 2019] s'intéressent aussi aux problèmes (3) et (4) en utilisant une modification de la méthode C-value pour recalculer les scores des termes-clés candidats.

Notre contribution par rapport aux autres est la résolution des quatre faiblesses dans une seule méthode. Pour cela, nous allons combiner les solutions apportées par Rui Wang *et al.* [Wang 2015] et Yeom *et al.* [Yeom 2019]. Cependant, nous utilisons le plongement de mots d'une manière plus simple que celles de Rui Wang *et al.* [Wang 2015] lors de la construction du graphe de mots. Pour la résolution des problèmes de scores associés aux termes-clés candidats, le chevauchement des termes-clés et la perte d'informations, nous adoptons les mêmes solutions que celles apportées par Yeom *et al.* [Yeom 2019]. Nous avons apporté notre modification à la méthode de Boudin *et al.* [Boudin 2013a] parce qu'elle utilise des algorithmes simples pour l'ordonnancement des nœuds du graphe et offre des performances équivalentes aux autres méthodes à base de graphes.

La suite de ce chapitre est organisée comme suit : la Section 2.2 présente l'état de l'art sur les méthodes d'extraction de termes-clés. La Section 2.3 détaille notre méthode d'extraction de termes-clés. La Section 2.4 présente les résultats de notre méthode. Enfin la Section 2.5 conclut ce chapitre.

2.2 Etat de l'art

Cet état de l'art comprend deux aspects : les méthodes de repérage et la sélection des termes-clés de la littérature et le plongement de mots qui est un élément clé de notre proposition.

2.2.1 Repérage et sélection des termes-clés

Les méthodes s'appuient sur deux étapes : le repérage ou choix des termes-clés candidats et la sélection parmi ces candidats des termes-clés à retenir. Dans la première étape, des méthodes linguistiques permettent de définir des termes-clés candidats. Dans la second étape, avec les méthodes supervisées, les termes-clés candidats sont classés en deux classes, soit dans la classe des termes-clés, soit dans celle des non termes-clés. Avec les méthodes non supervisées, les termes-clés candidats sont ordonnés selon un score d'importance et les meilleurs sont considérés comme des termes-clés.

2.2.1.1 Termes-clés candidats

L'objectif de la première phase est de déterminer les unités textuelles potentiellement des termes-clés : ce sont les termes-clés candidats. Ce sont des unités textuelles présentant des particularités similaires à celles des termes-clés fournis par des humains.

Cette étape du choix des termes-clés candidats offre deux avantages : l'une est la réduction du temps de calcul lors de l'extraction des termes-clés ; l'autre est la suppression des unités textuelles non pertinentes.

Cette étape est aussi importante pour la majorité des méthodes d'extraction automatique de termes-clés, supervisées ou non, parce que si les termes-clés du document analysé ne figurent pas dans l'ensemble des termes-clés candidats, alors ceux-ci ne pourront pas être retenus.

Il existe plusieurs méthodes de choix de termes-clés candidats, dans ses travaux, Hulth [Hulth 2003] distingue 3 méthodes d'extraction de termes candidats : *N-grammes*, *parties nominales* et *patrons grammaticaux prédéfinis*.

N-grammes. Les N-grammes sont les séquences ordonnées de **N** mots adjacents. Leur extraction est exhaustive et fourni un très grand nombre de termes-clés candidats sans qu'ils ne soient pour autant tous pertinents, c'est-à-dire plus vraisemblables de devenir des termes-clés. Afin de résoudre, en partie, des restrictions ont été proposées comme l'exclusion des N-grammes contenant des ponctuations et l'utilisation de la racinisation [Hulth 2003]. Une autre solution, couramment utilisée et efficace, est le filtrage avec un anti-dictionnaire regroupant les mots fonctionnels de la langue (prépositions, conjonctions, . . .) et les mots courants (presque, près, plusieurs, ...). Un N-gramme présentant, soit en début soit en fin, un des mots présents dans l'anti-dictionnaire ne sera pas sélectionné pour être un terme-clé candidat. Malgré son aspect très bruité, l'extraction de N-grammes est l'une des méthodes les plus utilisées parmi les méthodes supervisées [Turney 2000, Witten 1999] du fait de la simplicité de sa mise en œuvre et parce que la phase d'apprentissage des méthodes supervisées les rend moins sensibles aux éventuels bruits.

Illustration. Les 1 ⋯ 3-grammes sélectionnés (avec filtre) dans la phrase "Conceptualisation d'un système d'informations lexicales, une interface paramétrable pour le T.A.L." sont :

Uni-gramme	Bi-gramme	Tri-gramme
conceptualisation	informations lexicales	système d'informations
système	interface paramétrable	
informations		
lexicales		
interface		
paramétrable		
T.A.L		

Parties nominales. Les parties nominales ou "Np-chunks" sont des séquences contiguës et non-récursives d'unités linguistiques telles que la tête est un nom accompagné de ses éventuels déterminants et modifieurs. Ils sont linguistiquement définis, c'est-à-dire qu'ils sont plus fiables que les N-grammes, comme le prouvent les expériences réalisées par Hulth [Hulth 2003] consacrées à l'apport de connaissances linguistiques pour l'extraction automatique de termes-clés. En dépit de cela, Hulth remarque que l'utilisation de l'étiquetage grammatical sur les N-grammes permet d'obtenir de meilleures performances qu'avec les parties nominales.

Illustration. Les Parties nominales (ou Np-chunks) sélectionnées dans la phrase "Conceptualisation d'un système d'informations lexicales, une interface paramétrable pour le T.A.L." sont :

Parties nominales
conceptualisation
un système
informations lexicales
une interface paramétrable
le T.A.L

Patrons grammaticaux prédéfinis. Cette approche permet un contrôle avec précision de la nature et de la grammaticalité des termes-clés candidats à sélectionner. À l'instar des parties nominales, leur sélection est mieux fondée linguistiquement que celle des N-grammes. Hulth [Hulth 2003], dans ses travaux, sélectionne les termes-clés candidats à l'aide des patrons des termes-clés de référence qui sont les plus fréquents dans le corpus d'apprentissage (c'est-à-dire plus de dix occurrences), alors que les autres chercheurs, comme Wan et Xiao [Wan 2008] et Hasan et Ng [Hasan 2010],

se focalisent seulement sur les plus longues séquences de noms (noms propres inclus) et d'adjectifs.

Illustration. Les plus longues séquences de noms et d'adjectifs sélectionnés dans la phrase "Conceptualisation d'un système d'informations lexicales, une interface paramétrable pour le T.A.L" sont :

Patrons grammaticaux
conceptualisation
système
informations lexicales
interface paramétrable
T.A.L

2.2.1.2 Méthodes d'extraction de termes-clés

Les méthodes d'extraction de termes-clés peuvent être classées en deux : les méthodes supervisées et les méthodes non supervisées.

Les méthodes d'extraction automatique de termes-clés effectuent soit un ordonnancement des termes-clés candidats (majorité des méthodes non supervisées), soit une classification des termes-clés candidats entre les classes "terme-clé" et "non terme-clé" (majorité des méthodes supervisées). Les méthodes supervisées requièrent des documents préalablement annotés par les termes-clés. Ce sont les termes-clés fournis par des auteurs qui sont généralement utilisés pour l'entraînement des modèles de classification. Cette utilisation de l'apprentissage constitue leur grande faiblesse parce que la précision de ces méthodes dépend grandement du corpus d'apprentissage.

Les méthodes non supervisées quant à elles n'ont pas besoin de corpus d'apprentissage, ce qui leur permet de s'abstraire du domaine disciplinaire et de la langue. Elles exploitent directement différentes caractéristiques (par exemple les traits statistiques) des termes-clés candidats dans le document ou dans le corpus.

Dans la suite, nous aborderons d'abord les méthodes supervisées d'extraction automatique de termes-clés puis les méthodes non supervisées, plus particulièrement celles à base de graphes. Enfin, nous dresserons un bilan sur les méthodes d'extraction automatiques de termes-clés.

Méthodes supervisées

Avec les méthodes supervisées, le problème d'extraction de termes-clés est souvent considéré comme un problème de classification. Ces méthodes apprennent à prédire

si un terme-clé candidat peut être classé en tant que "terme-clé" ou non. Compte tenu de l'évolution de l'apprentissage supervisé, les méthodes supervisées peuvent aussi réaliser un ordonnancement des termes-clés candidats.

Quant on parle de méthodes supervisées, un corpus d'apprentissage ou d'entraînement est toujours nécessaire. Pour l'extraction de termes-clés, le corpus d'apprentissage est composé de documents annotés par des termes-clés (ce sont les exemples) et par des non termes-clés (contre-exemples). Les exemples sont les termes-clés des auteurs ou ceux manuellement donnés par d'autres personnes, tandis que les contre-exemples sont les termes-clés candidats sélectionnés dans les documents qui ne font pas partis des exemples ou termes-clés. Le principe général de ces méthodes est de trouver (calculer) les caractéristiques qui représentent les termes (termes-clés ou non) dans le corpus d'apprentissage comme la fréquence, l'ordre d'apparition dans le document, etc. puis les utiliser pour l'apprentissage des classifieurs. Ensuite, pour un nouveau document qui n'est pas dans le corpus d'apprentissage, ses termes-clés candidats sont aussi représentés avec les mêmes caractéristiques utilisées durant l'apprentissage puis les classifieurs entraînés distinguent si un candidat est un terme-clé ou non.

Dans cette section, nous allons présenter les méthodes supervisées selon le type de classifieurs utilisés.

Classifieurs probabilistes.

Le principe des classifieurs probabilistes est de combiner les distributions de probabilités de divers caractéristiques des termes-clés candidats afin de calculer le score de vraisemblance, la probabilité pour que ces candidats deviennent des termes-clés. Ces probabilités peuvent être utilisées pour ordonner les termes-clés dans le cas où il ne faut extraire qu'un nombre donné de termes-clés comme dans les méthodes non supervisées.

L'une des méthodes la plus populaire dans cette catégorie est celle proposée par Witten *et al.* [Witten 1999] nommée KEA et qui a inspiré plusieurs autres méthodes. L'hypothèse de la méthodes KEA est qu'un terme-clé est important vis-à-vis de sa fréquence dans le corpus ainsi que de son ordre d'apparition dans les différentes parties des documents du corpus (par exemple le titre, le résumé, etc.). À partir de cette hypothèse, la méthode définie deux caractéristiques des termes-clés candidats qui sont : leur première position dans le document analysé et leur poids Tf-Idf. KEA combine alors les distributions probabilistes de ces deux caractéristiques et effectue une classification naïve bayésienne pour calculer le score de vraisemblance des termes-clés candidats. Frank *et al.* [Frank 1999] améliore KEA en utilisant une caractéristique supplémentaire le nombre de fois qu'un terme-clé candidat est un exemple.

Turney *et al.* [Turney 2003] modifie aussi KEA en ajoutant une deuxième classification naïve bayésienne après celle de KEA. L'hypothèse des auteurs est d'améliorer la cohérence entre les termes-clés candidats extraits. En effet, la première classification permet d'ordonner les candidats selon leur score de vraisemblance puis la deuxième classification attribue un meilleur score de vraisemblance aux termes-clés candidats ayant un fort lien sémantique avec un ou plusieurs candidat(s).

Comme une autre amélioration apportée à KEA, Nguyen *et al.* [Nguyen 2007] proposent de nouvelles caractéristiques pour l'extraction des termes-clés dans les articles scientifiques. Leur hypothèse est qu'il n'y a pas de répartition homogène des termes-clés dans les différentes sections d'une publication scientifique. Les auteurs ont donc ajouté aux caractéristiques de KEA, des caractéristiques qui capturent les positions des termes-clés par rapport aux sections du document et des caractéristiques morphologiques comme savoir si le terme-clé est une abréviation ou non.

Dans un autre travail, Nguyen *et al.* [Nguyen 2010] ont également amélioré KEA en ajoutant 17 autres caractéristiques à celles de KEA, dont celles de leur précédent travail ([Nguyen 2007]) qui capturent les positions des termes-clés par rapport aux sections du document. Ils ont testé différentes combinaisons de ces caractéristiques ainsi que différentes longueurs du document, en n'utilisant que quelques parties du document (combinaison de différentes sections). Ils ont nommé ce dernier travail WINGNUS.

Caragea *et al.* [Caragea 2014] proposent CeKE, un modèle qui utilise un classifieur naïf bayésien. Ce modèle utilise trois caractéristiques déjà existantes, trois caractéristiques existantes améliorées et trois nouvelles caractéristiques qu'ils ont proposées : un score Tf-Idf de chaque candidat calculé à partir des contextes citationnels, ainsi que deux caractéristiques binaires indiquant si le candidat apparaît dans (1) une phrase (du document qui contient le candidat) qui cite un autre document ou (2) une phrase d'un autre document qui cite le document. Ce type de modèle présente un défaut qui est la difficulté d'extraire les termes-clés des documents qui ne sont pas cités par d'autres articles.

Gollapalli *et al.* [Gollapalli 2017] utilisent Conditional Random Fields (CRF) comme classifieur pour la tâche d'extraction de termes-clés. Ce classifieur a la capacité d'étiqueter tous les mots du document en donnant les classes suivantes : **KP** correspond à un mot d'un terme-clé et **O** fait référence à un mot d'un non terme-clé. Gollapalli *et al.* utilisent des caractéristiques représentant les informations linguistiques, orthographiques et structurelles du document. Ils ont aussi étudié les caractéristiques. Ils ont aussi étudié l'incorporation de connaissance externe dans leur processus d'extraction de termes-clés.

Récemment, Florescu *et al.* [Florescu 2018] proposent SurfKE, un modèle qui utilise

un classifieur naïf bayésien Gaussien. Leur approche consiste à construire un graphe de mots représentant le document puis à extraire les caractéristiques à partir du graphe de mots.

Séparateurs à Vaste Marge.

Parmi les premiers à utiliser les SVM, il y a Zhang *et al.* [Zhang 2006]. Comme caractéristiques pour représenter les termes, ils ont utilisé le contexte global et le contexte local. Le contexte global d'un terme-clé candidat est représenté par son Tf-Idf, sa première position et ses occurrences dans différentes parties du document. Le contexte local d'un terme-clé est représenté par sa catégorie grammaticale, le nombre de fois qu'il modifie un mot, le nombre de fois qu'un mot le modifie et la somme du Tf-Idf de tous les mots qui co-occurrent avec le candidat. Cette dernière caractéristique indique si un terme-clé candidat apparaît dans un contexte important vis-à-vis du document.

Il y aussi un type particulier de SVM qui permet d'ordonner les termes-clés candidats en construisant plusieurs plan de projections. Ce dernier est nommé SVM^{rank}. L'un des travaux qui utilise ce type de SVM est celui de Jiang *et al.* [Jiang 2009] qui ont montré qu'en utilisant les mêmes caractéristiques (Tf-Idf, taille en nombre de mots, etc.) des termes-clés candidats, SVM^{rank} fournit de meilleurs résultats qu'un SVM classique ou qu'un classifieur naïf bayesien.

Utilisant aussi SVM^{rank}, Eichler *et al.* [Eichler 2010] font l'hypothèse qu'un terme-clé candidat fréquemment utilisé comme terme-clé dans le corpus d'apprentissage est plus vraisemblablement un terme-clé. Alors durant l'entraînement du classifieur, ils ont assigné des poids aux exemples et contre-exemples de chaque document du corpus d'apprentissage : un poids égal à 2 pour les exemples, égal à 1 pour les contre-exemples du document mais qui sont des exemples d'autres documents dans le corpus et égal à 0 pour les autres contre-exemples. Ils ont aussi utilisé plusieurs caractéristiques des termes-clés candidats, qui sont des très utilisées, comme le poids Tf-Idf, la position dans le document, etc. En plus de ces caractéristiques, ils ont défini une nouvelle caractéristique qui se réfère à Wikipédia et indique si le candidat fait l'objet d'un article Wikipédia. Selon leur hypothèse, un terme-clé candidat est plus vraisemblablement un terme-clé s'il fait l'objet d'un article Wikipédia.

Réseaux de neurones.

Dans cette section, nous allons considérer des perceptrons multicouches qui sont des classifieurs inspirés du fonctionnement biologique de l'apprentissage humain. Les perceptrons multicouches sont des réseaux de neurones comprenant au moins trois couches dont chaque couche est composée de neurones. La première couche représente les caractéristiques d'un terme-clé candidat (un neurone représente une caracté-

ristique), les couches intermédiaires ou couches cachées propagent les scores obtenus selon la valeur des caractéristiques, et la dernière couche fournit le score final pour chaque classe "terme-clé" et "non terme-clé".

Les travaux de Sarkar *et al.* [Sarkar 2010] font parti de ceux qui utilisent un perceptron multicouche. Comme caractéristiques des termes-clés candidats, ils ont choisi la fréquence du candidat, sa position dans le document, sa taille (nombre de mots le composant) et la tailles des mots qui le composent (c'est-à-dire le nombre de caractères). L'intuition derrière le choix de ces caractéristiques est que la taille d'un mot est une indication de sa spécificité vis-à-vis du document car les mots courts sont plus fréquents que les mots longs (loi de Zipf (1935)). En utilisant le classifieur perceptron multicouche, ils obtiennent un score pour chaque classe et le terme-clé candidat est attribué à la classe ayant le score le plus élevé. Ces scores peuvent aussi être utilisés pour déterminer le degré de confiance attribué aux termes-clés candidats [Denker 1990]. Ces scores de confiance sont utilisés par Sarkar *et al.* [Sarkar 2010] pour connaître l'importance des termes-clés candidats en les ordonnant et permet d'extraire les top N termes-clés. Pour cela, ils ont placés les termes-clés candidats classés comme "termes-clés" dans l'ordre décroissant du score de confiance puis suivent ceux classés comme "non termes-clés" dans l'ordre croissant du score de confiance. Ainsi, ce sont les candidats classés comme "termes-clés" qui seront extraits en premier dans les top N.

Arbres de décision.

Les arbres de décision sont des classifieurs qui représentent un ensemble de choix sous la forme d'un arbre tels que les branches représentent des tests sur des caractéristiques des termes-clés candidats. Ces tests indiquent la branche à suivre jusqu'aux feuilles de l'arbre ; ces feuilles représentent les classes "terme-clé" ou "non terme-clé".

Dans ses travaux, Turney [Turney 2000] utilise l'arbre de décision C4.5 avec différentes configurations pour extraire les termes-clés d'un document. Il a conclu qu'en utilisant une forêt d'arbres C4.5 et en réduisant l'extraction de termes-clés à un vote, il obtient de bons résultats. C'est-à-dire que chaque arbre de décision C4.5 classe indépendamment les termes-clés candidats et ceux qui sont classés majoritairement comme "terme-clé" par tous les arbres sont considérés comme termes-clés. Comme valeur d'entrée utilisée aux arbres de décision, Turney utilise 9 caractéristiques pour représenter un terme-clé candidat dont 3 d'entre elles sont des catégories grammaticales précises telles que les adjectifs, et les 6 autres sont des statistiques standards comme le nombre de mots.

Ercan *et al.* [Ercan 2007] utilisent aussi l'arbre de décision C4.5 pour extraire les mots-clés. Dans leurs travaux, ils se sont restreints aux mots et n'ont pas proposé

d'extraire les termes-clés. Ils ont ajouté aux caractéristiques classiques (première position d'occurrence, nombre d'occurrences des mots et dernière position d'occurrence), 4 nouvelles caractéristiques basées sur les chaînes lexicales. Une chaîne lexicale est un graphe qui lie hiérarchiquement les mots d'un document en tenant compte du type de la relation. Ercan *et al.* [Ercan 2007] ont considéré comme relation : la méronymie, l'hyperonymie/hyponymie et la synonymie. A chaque relation est attribué un poids puis pour chaque mot du document, les 4 caractéristiques sont calculées comme suit : la première est la somme du poids de toutes les relations de la chaîne lexicale, la seconde est la somme du poids de toutes les relations du mot avec les autres mots de la chaîne lexicale, la troisième est la différence entre la position de la dernière occurrence d'un mot de la chaîne lexicale avec la position de la première occurrence d'un mot de la chaîne lexicale, et la dernière caractéristique est calculée de la même façon que la troisième, mais en considérant uniquement le mot concerné et ses voisins dans la chaîne.

Lopez *et al.* [Lopez 2010], vainqueurs de la campagne d'évaluation Semeval 2010 [Kim 2010], utilisent aussi les arbres de décision. Leur approche d'extraction de termes-clés se fait en 2 étapes : (a) classification des termes-clés candidats avec les arbres de décision, et (b) ordonnancement des termes-clés candidats. Pour réaliser l'ordonnancement, ils ont assigné un nouveau score à tous les termes-clés candidats. Ce nouveau score pour un candidat est calculé en fonction de son ancien score (score de confidence produit par la forêt d'arbres de décision comme avec les réseaux de neurones) et des scores des autres termes-clés candidats. Ce dernier ordonnancement permet de capturer les relations entre les termes-clés candidats. Ils ont utilisé plusieurs caractéristiques allant de simples statistiques comme le nombre de mots, la position dans le document, à des caractéristiques plus complexes comme la mesure de la cohésion lexicale d'une séquence de mots dans un document donné.

Medelyan *et. al* [Medelyan 2009], quant à eux, utilisent un ensemble d'arbres de décision (bagged decision trees) pour la tâche d'extraction automatique de termes-clés. Ils ont utilisé plusieurs caractéristiques dans leur approche comme le poids Tf-Idf, la première position dans le document, un score qui quantifie combien de fois le candidat apparaît comme terme-clé dans le corpus d'entraînement, etc. Ils ont nommé leur approche Maui.

Apprentissage profond.

Les apprentissages profonds qui sont énormément utilisés dans les traitements et classifications d'images sont aussi beaucoup utilisés dans les traitements de textes. Dans cette section, nous allons voir les méthodes d'extraction automatique de termes-

clés qui utilisent les apprentissages profonds.

Zhang *et al.* [Zhang 2016] proposent un modèle basé sur un Recurrent Neural Networks (réseaux de neurones récurrents) (RNN) profond pour la tâche d'extraction de termes-clés pour les tweets. Leur modèle traite conjointement l'ordonnancement de mots-clés, la génération de termes-clés à partir des mots-clés et l'ordonnancement de termes-clés. Le modèle possède deux couches cachées pour discriminer les mots-clés et classer les termes-clés, et ces deux sous-objectifs sont combinés en une fonction d'objectif final.

Meng *et al.* [Meng 2017] eux proposent un modèle génératif basé sur l'architecture encodeur-décodeur pour extraire les termes-clés des textes scientifiques. Plus précisément, ils utilisent un modèle RNN encodeur-décodeur qui est capable d'extraire des termes-clés qui apparaissent rarement dans le document, mais aussi de générer ceux qui n'y apparaissent pas. Cependant, leur approche ignore les liens potentiels entre les termes-clés, ce qui peut conduire à ce que les termes-clés ne couvrent pas tous les thèmes du document ou à l'inverse que deux termes-clés soient redondants. Chen *et al.* [Chen 2018] résolvent ces deux problèmes en tenant compte des corrélations entre les termes-clés et en ajoutant deux contraintes : les termes-clés doivent couvrir tous les thèmes et doivent être différents les uns des autres.

Ye *et al.* [Ye 2018], quant à eux, proposent des méthodes semi-supervisées de génération de termes-clés basées aussi sur un modèle encodeur-décodeur. Ils essaient de résoudre le problème des méthodes supervisées de [Meng 2017, Chen 2018] qui nécessitent des quantités massives de documents étiquetés avec des termes-clés pour le processus d'entraînement. Pour cela, Ye *et al.* proposent deux approches : la première consiste à associer aux documents non étiquetés des termes-clés extraits avec des méthodes non supervisées puis combiner ce document avec ceux déjà manuellement annotés pour former le corpus d'entraînement. La seconde approche consiste à faire un apprentissage multi-tâche, c'est-à-dire combiner la tâche de génération de termes-clés avec une tâche de génération de titre [Rush 2015]. Les deux tâches partagent le même réseau d'encodeurs, mais ont différents décodeurs. Durant le processus d'entraînement, les documents non étiquetés sont utilisés pour estimer les paramètres du réseau d'encodeurs avec la tâche de génération de titre puis ce même réseau d'encodeur est ensuite utilisé comme encodeur du modèle de génération de termes-clés. Enfin, le modèle de génération de termes-clés est ré-entraîné avec des documents étiquetés pour obtenir le modèle final.

Wang *et al.* [Wang 2018] résolvent aussi le problème d'insuffisance de documents étiquetés dans les données d'entraînement. Pour cela, ils proposent Topic-based Adversarial Neural Network (TANN), un modèle qui peut être entraîné avec un corpus com-

posé de documents étiquetés et non-étiquetés. Pour cela, le modèle utilise des connaissances acquises depuis une autre source riche en ressources (documents étiquetés) pour l'extraction de termes-clés des documents non étiquetés.

Basaldella *et al.* [Basaldella 2018] eux proposent un modèle Bidirectional Long Short-TermMemory (Bi-LSTM). Le modèle attribut à chaque mot du document une des trois étiquettes suivantes : non mot-clé, début d'un terme-clé et composant d'un terme-clé. À partir de ces mots étiquetés les termes-clés sont générés. Alzaidy *et al.* [Alzaidy 2019] quant à eux proposent un modèle qui combine Bi-LSTM et CRF. Dans leur modèle, la couche Bi-LSTM sert à capturer la sémantique puis la couche CRF utilise les informations sémantiques pour attribuer à chaque mot du document l'une des étiquettes suivantes : mot d'un terme-clé (KP) et non mot d'un terme-clé (Non-KP). Les termes-clés sont ensuite générés à partir des mots qui sont étiquetés comme KP.

Autres approches.

Bougouin *et al.* [Bougouin 2016] proposent une méthode supervisée à base de graphes, qui est une extension d'une méthode non supervisée TopicRank [Bougouin 2013b]. Cette nouvelle méthode, appelée TopicCoRank, consiste à créer deux graphes : un graphe de sujets construit à partir du document où il faut extraire les termes-clés, un graphe de termes-clés contrôlés construit à partir des termes-clés des documents du corpus d'entraînement. Une stratégie a été conçue pour unifier les deux graphes puis à l'aide d'un vote de co-classement, les sujets et termes-clés contrôlés sont ordonnés. Puis les termes-clés du document sont les N sujets et termes-clés contrôlés les mieux classés. Comme un sujet est un ensemble de candidats, la même stratégie que dans TopicRank a été adoptée pour choisir un candidat pour représenter le sujet et ce candidat sera considérer comme terme-clé.

Yang *et al.* [Yang 2018] proposent une méthode pour extraire les termes-clés dans les média sociaux ; chaque terme-clé correspond à un évènement particulier. Leur méthode se divise en trois étapes : extraction de termes dans le documents, extraction de termes-clés candidats associés à des événements spécifiques en utilisant Task-oriented Latent Dirichlet Allocation model (ToLDA), et enfin l'algorithme PMI-IR [Turney 2001] est utilisé pour obtenir les synonymes des termes-clés candidats puis tous les candidats, les synonymes inclus, sont évalués par des personnes et les candidats sélectionnés sont considérés comme termes-clés.

Wan *et al.* [Wang 2017] eux proposent d'utiliser un ensemble de méthodes supervisées et non supervisées pour l'extraction de termes-clés. Ils utilisent des caractéristiques des termes-clés candidats extraits du document comme le poids Tf-Idf, ainsi que des caractéristiques issues de connaissances externes comme Wikipédia.

Ensuite, ils ont utilisé un ensemble de deux méthodes non supervisées (TextRank [Mihalcea 2004] et SGRank [Danesh 2015]) et deux classifieurs (un forêt d'arbre aléatoire et un Séparateurs à Vaste Marge (SVM) linaire) pour ordonner les termes-clés candidats.

Méthodes non supervisées

La majorité des méthodes non supervisées d'extraction de termes-clés sont formulées comme un problème d'ordonnancement, il s'agit d'ordonner les termes-clés candidats en leur donnant un score d'importance vis-à-vis du contenu du document, puis de considérer les k plus importants (ceux qui ont les meilleurs scores) en tant que termes-clés. Ces méthodes peuvent s'abstraire du domaine des documents qu'ils traitent et peuvent donc être appliquées dans plusieurs situations. Dans cette section, nous présentons différentes méthodes classées dans cette catégorie.

Méthodes statistiques.

Les méthodes statistiques utilisent majoritairement des indicateurs statistiques pour quantifier l'importance des termes-clés candidats comme le nombre d'occurrences soit dans le document, soit dans un corpus de référence, ou bien dans les deux.

Les méthodes Term Frequency-Inverse Document Frequency (Tf-Idf) [Salton 1975] et Likey [Paukkeri 2010] mesurent l'importance d'un terme dans un document donné relativement au corpus. L'hypothèse de ces méthodes est qu'un terme devient d'autant plus important dans un document qu'il y apparaît à de nombreuses reprises et qu'il est moins fréquent dans les autres documents du corpus.

$$Tf - Idf\,(terme) = Tf\,(terme) \times \log\left(\frac{N}{Df\,(terme)}\right) \tag{2.1}$$

$$Likey\,(terme) = \frac{rang_{document}\,(terme)}{rang_{corpus}\,(terme)} \tag{2.2}$$

où :
Tf : nombre d'occurrences d'un terme dans le document analysé
Df : nombre de documents dans le corpus dans lequel le terme est présent
N : nombre total de documents
$rang_{document}\,(terme)$ **:** nombre d'occurrences d'un terme dans le document
$rang_{corpus}\,(terme)$ **:** nombre d'occurrences d'un terme dans le corpus.

Plus la valeur Tf-Idf d'un terme-clé candidat est élevée, plus celui-ci est important dans le document analysé. Et inversement, plus la valeur Likey d'un terme-clé candidat est faible, plus celui-ci est important dans le document analysé.

La méthode Tf-Idf est à ce jour l'une des méthodes de référence parmi les méthodes non supervisées d'extraction automatique de termes-clés et est souvent utilisée comme méthode de référence dans les comparaisons.

El-Beltagy *et al.* [El-Beltagy 2010] proposent une méthode d'extraction de termes-clés appelée KP-Miner. Cette méthode filtre d'abord les termes-clés candidats en supprimant ceux qui apparaissent moins de 3 fois et ceux qui apparaissent pour la première fois au-delà d'une certaine position dans le document. Puis, pour ordonner les candidats restant, ils ont utilisé une formule transformée de Tf-Idf qui prend en compte la longueur du document.

Campos *et al.* [Campos 2018] proposent YAKE, une méthode d'extraction automatique de termes-clés non supervisée utilisant cinq caractéristiques statistiques calculées à partir de textes comme la fréquence de mots. Ces caractéristiques sont combinées pour former une seule mesure (qui sera considérée comme le score du mot) et sont calculées pour tous les mots du document (excepté les mots vides). Puis les auteurs ont proposé une formule pour calculer le score d'un terme-clé candidat en fonction des scores et des fréquences des mots qui le composent. Ensuite, ils éliminent les termes-clés candidats similaires (ceci est équivalent à supprimer les doublons). La similarité est calculée en utilisant la distance de Levenshtein. Finalement, ils obtiennent une liste de termes-clés ordonnés selon leurs scores.

Won *et al.* [Won 2019] proposent comme approche une combinaison de quatre caractéristiques statistiques simples calculées à partir de textes comme la fréquence, la longueur d'un terme-clé candidat. Le score final de chaque terme-clé candidat est le résultat du produit de ces caractéristiques statistiques et finalement les top-N termes-clés candidats sont considérés comme termes-clés.

Approches par regroupement.

Les méthodes par regroupement définissent ou créent des groupes d'unités textuelles partageant au moins une caractéristique commune comme la similarité lexicale, la similarité sémantique, etc. Par conséquent, les termes-clés extraits dans chaque groupe couvrent mieux le document analysé selon les caractéristiques utilisées [Bougouin 2013a].

Matsuo *et al.* [Matsuo 2004] proposent une méthode qui ne regroupe que les termes les plus fréquents en utilisant un lien sémantique (de co-occurrence) entre les termes. Plus précisément, ils sélectionnent les 30 % des termes-clés candidats les plus fréquents,

puis groupent ceux qui co-occurrent fréquemment dans une phrase. Le regroupement terminé, ils comparent les groupes de termes fréquents avec les termes-clés candidats du document, avec l'hypothèse qu'un terme-clé candidat ayant une co-occurrence élevée avec les termes fréquents d'un ou plusieurs groupes, est plus vraisemblablement un terme-clé.

Dans leur algorithme KeyCluster, Liu *et al.* [Liu 2009] emploient aussi un regroupement sémantique, en ne prenant en compte que les mots du document analysé et non les mots outils issus d'un anti-dictionnaire par exemple. Les groupes sémantiques sont chacun représentés par un mot de référence, qui est aussi le mot le plus central de chaque groupe. Les termes-clés sont tous les termes-clés candidats qui contiennent au moins un mot de référence. L'avantage de cette méthode est qu'elle offre une bonne couverture des thèmes abordés dans le document, du fait que tous les groupes sémantiques sont représentés par au moins un terme-clé. Mais elle présente aussi l'inconvénient de ne pas pondérer les termes-clés, par conséquent aucun ordonnancement ne peut être fait pour déterminer les termes-clés les plus importants.

Approches à base de graphes.

Les méthodes à base de graphes font partie des méthodes les plus populaires actuellement. En effet, les graphes peuvent présenter de manière simple et intuitive un document textuel. Ces méthodes se décomposent généralement en trois phases : sélection des termes-clés candidats, la construction du graphe de termes qui représente le document analysé, l'ordonnancement des termes-clés candidats et la sélection ou extraction des termes-clés.

Mihalcea *et al.* [Mihalcea 2004] proposent une approche appelée TextRank, qui est une méthode inspirée de PageRank [Brin 1998], un algorithme de marche aléatoire d'ordonnancement de page Web. TextRank ordonne les unités textuelles à partir d'un graphe pour extraire les termes-clés ou/et pour faire un résumé automatique d'un document. Pour l'extraction des termes-clés, chaque nœud du graphe est un mot du document et l'arête qui connecte deux nœuds représente leur relation d'adjacence dans le document. Ainsi, une arête existe si les deux mots co-occurrent dans une fenêtre de deux mots. Ensuite, pour chaque nœud, un score d'importance (initialement égal à un) est calculé à l'aide de l'algorithme itératif PageRank [Brin 1998]. PageRank va parcourir le graphe de mot en mot en se déplaçant vers un mot qui co-occurre avec le mot courant. L'importance de chaque mot est déduite d'après le principe du vote, c'est-à-dire qu'un mot est important s'il co-occurre avec un grand nombre de mots et si les mots avec lesquels il co-occurre sont eux aussi importants. Enfin, les termes-clés sont les plus longues séquences des mots les plus importants dans le document.

Wan *et al.* [Wan 2008] proposent la méthode SingleRank qui est une proposition d'amélioration de TextRank. Leur hypothèse est d'augmenter la précision de l'ordonnancement en élargissant la fenêtre de co-occurrence à dix mots et en pondérant les arêtes par le nombre de co-occurrences entre les deux mots. La pondération permet d'ajuster l'importance des mots à partir de ses recommandations par les autres mots qui lui sont reliés. Les auteurs ont aussi proposé d'ordonner les termes-clés à partir de la somme du score d'importance des mots qui les composent au lieu de les générer à partir des séquences de mots importants dans le document. Comparé à TextRank, SingleRank donne de meilleurs résultats d'après les expériences menées par Hasan *et al.* [Hasan 2010] sur quatre collections de données différentes.

Wan *et al.* [Wan 2008] proposent une autre méthode appelée ExpandRank qui est une extension de SingleRank. ExpandRank suit l'hypothèse que des documents similaires au document traité (d'après la mesure de similarité vectorielle cosinus) fournissent des informations supplémentaires relatives aux mots du document et aux relations qu'ils entretiennent. Cette nouvelle méthode ajoute et renforce des arêtes dans le graphe de mots en utilisant les relations de co-occurrences observées dans les documents similaires.

Liu *et al.* [Liu 2010] ont aussi proposé une amélioration de SingleRank qu'ils ont nommé Topical PageRank (TPR). Cette méthode cherche à augmenter la couverture du document par les termes-clés qu'elle extraits. En premier, ils ont entraîné un modèle Latent Dirichilet Allocation (LDA) [Blei 2003] pour connaître les thèmes abordés dans le document. Ensuite, ils ont construit un graphe de mots à partir du document (comme avec SingleRank), puis ils calculent l'importance de chaque mot du graphe pour chaque thème abordé. Le score d'un mot pour un thème donné est obtenu en intégrant à son score PageRank la probabilité qu'il appartienne à ce thème. Finalement, le score global d'un terme-clé candidat est obtenu en fusionnant ses scores pour chaque thème en sachant que le score d'un candidat est la somme des scores des mots qui le composent. Sterckx *et al.* [Sterckx 2015a] proposent une version améliorée de TPR qu'ils ont appelé Single Topical PageRank (STPR). STPR propose de réduire le nombre de calculs tout en conservant l'intuition de recouvrement de thème en exécutant un seul PageRank pour chaque document. Dans une autre étude, Sterckx *et al.* [Sterckx 2015b] proposent de combiner plusieurs modèles de thèmes. Ils ont montré que la moyenne de plusieurs modèles de thèmes entraîne une augmentation de la précision des termes-clés extraits.

Dans le même type de méthode que TPR et STPR, Teneva *et al.* [Teneva 2017] proposent SalienceRank. Cette méthode, comme STPR, exécute un seul PageRank et y incorpore une nouvelle mesure appelée "saillance". La saillance d'un mot est une combinaison linéaire de deux mesures : la spécificité du thème (mesure combien le mot est

partagé entre les thèmes) et la spécificité du corpus (mesure la fréquence du mot dans le corpus).

Florescu *et al.* [Florescu 2017] ont aussi proposé une méthode basée sur l'algorithme PageRank appelée PositionRank. Les étapes sont les mêmes que celles de la méthode SingleRank de Wan *et al.* [Wan 2008]. Les auteurs ont simplement modifié PageRank (qui calcule les scores des mots) en tenant compte de toutes les positions des occurrences des mots dans le document. En effet, leur hypothèse est que les termes-clés apparaissent à des positions très proches du début d'un document, même dans le titre, et ils sont fréquents.

Boudin *et al.* [Boudin 2013a] comparent différentes mesures de centralité appliquées au graphe de mots (comme dans SingleRank, mais seuls les noms et les adjectifs sont retenus) pour calculer le score de chaque mot, puis ils obtiennent les scores des termes-clés candidats en sommant les scores des mots qui les composent. Ils ont conduit des expériences sur des ensembles de données en anglais et en français et ont constaté qu'en utilisant la simple mesure de centralité "degré" ils obtiennent des résultats comparables à l'algorithme TextRank. La mesure de centralité de proximité permet quant à elle d'obtenir les meilleurs résultats sur des documents courts.

Dans un autre article de leur travail, Boudin *et al.* [Bougouin 2013b] proposent une nouvelle approche appelée TopicRank. TopicRank se démarque des autres méthodes précédentes en proposant de traiter directement les termes-clés candidats et non les mots qui les composent. Cette méthode se déroule donc en quatre étapes : l'identification des sujets, la construction du graphe des sujets, l'ordonnancement des sujets, la sélection des termes-clés. Un sujet représente un concept véhiculé par une ou plusieurs unités textuelles, autrement dit un sujet est un groupe de termes-clés candidats qui sont similaires (similarité de Jaccard). Le groupement des termes-clés candidats en sujets est effectué avec l'algorithme de groupement hiérarchique agglomératif et une fois les sujets définis, le graphe de sujets, qui est un graphe complet, est construit. Le poids de l'arête entre deux nœuds (sujets) est quantifié à partir des relations des mots des termes-clés candidats qui les composent. Les sujets sont ensuite ordonnancés en appliquant l'algorithme PageRank. Chaque sujet important va fournir un terme-clé et pour choisir parmi les termes-clés candidats qui le composent, il existe trois moyens : *Position* (le candidat qui apparaît en premier dans le document), *Fréquence* (le candidat le plus fréquent), et *Centroïde* (le candidat le plus similaire aux autres candidats).

Boudin [Boudin 2018] propose également une modification de TopicRank. La première différence est observée lors de la construction du graphe. Dans TopicRank, un graphe de sujets, non orienté, est construit alors qu'avec cette méthode, c'est un graphe orienté de termes-clés candidats tels que les candidats du même sujet ne sont pas re-

liés. Le poids de l'arête entre deux candidats est calculé comme la somme des distances inverses entre les occurrences de ces candidats. Après que le graphe soit construit, un ajustement des poids des arêtes est réalisé en considérant les positions des termes-clés candidats dans le document. Plus précisément, les poids des arêtes entrant vers les candidats qui apparaissent au début du document sont augmentés en utilisant les poids des arêtes sortant des autres candidats du même sujet. La seconde différence se passe lors de l'étape d'ordonnancement : avec cette nouvelle méthode, ce sont les termes-clés candidats qui sont ordonnés et non les sujets, puis c'est la méthode TextRank qui a été utilisée. Enfin, comme ce sont les candidats qui sont ordonnés, il n'est plus besoin de choisir les représentants des sujets comme dans TopicRank, les candidats avec les meilleurs scores seront considérés comme termes-clés.

Rose *et al.* [Rose 2010] eux proposent une méthode appelée Rapid Automatic Keyword Extraction (RAKE). RAKE utilise une liste de mots vides, un ensemble de délimiteurs de termes et un ensemble de délimiteurs de mots pour partitionner le texte en termes-clés candidats. Ensuite avec le graphe de mots, construit comme dans TextRank, un score pour chaque nœuds est calculé avec l'une des trois méthodes suivantes : la fréquence du mot dans le document ou le degré du mot dans le graphe ou le rapport degré/fréquence. Enfin le score d'un terme-clé candidat est la somme des scores des mots qu'il contient et les N premiers candidats sont considérés comme termes-clés.

Danesh *et al.* [Danesh 2015] proposent une méthode appelée SGRank qui combine les méthodes statistiques et à base de graphes. Premièrement, SGRank extrait les termes-clés candidats du document puis attribue un score calculé en utilisant une version modifiée de Tf-Idf. Ensuite, les scores des candidats sont recalculés en utilisant des statistiques heuristiques comme la longueur du candidat ou la position de la première occurrence dans le document. Enfin, ces candidats et leurs scores sont incorporés dans un algorithme à base de graphes pour ordonner les candidats et les N premiers sont considérés comme termes-clés.

Gollapalli *et al.* [Gollapalli 2014] proposent CiteTextRank, une méthode qui utilise, pour le processus d'extraction de termes-clés, un algorithme à base de graphes et qui incorpore des informations provenant à la fois du contenu du document et de réseaux de citations.

Approches basées sur le plongement de mots.

Rui Wang *et al.* [Wang 2015] utilisent le plongement de mots (voir Section 2.2.2 pour plus de détails sur ce principe) comme connaissance d'arrière-plan pour créer un graphe de mots, puis les auteurs utilisent un algorithme PageRank pondéré pour calculer le score de chaque nœud afin de les classer. Plus précisément, les auteurs ont

construit un graphe de mots tel que les sommets sont seulement des noms et des adjectifs. Puis, deux nœuds ou mots w_i et w_j sont liés s'ils co-occurrent dans une fenêtre de C mots ($C = 2, 5\ et\ 10$ ont été testés) et ils ont utilisé le plongement de mots pour calculer le poids des arêtes. Les auteurs ont proposé 4 différents moyens de calculer le poids d'une arête qui sont des combinaisons de mesures statistiques (exemple : nombre de co-occurrences, fréquence de mots, ...) et de mesures de distances entre deux vecteurs obtenus par le plongement de mots (distance cosinus et distance euclidienne). Après avoir utilisé l'algorithme de PageRank pondéré sur le graphe de mots construits, ils ont calculé les scores des termes-clés candidats, qui sont de longues séquences de noms et adjectifs se terminant uniquement par des noms, en sommant les scores des mots qui les composent.

Mahata *et al.* [Mahata 2018] proposent Key2vec, une méthode non supervisée d'extraction de termes-clés à partir des documents scientifiques. Leur approche se déroule comme suit : premièrement, les termes-clés candidats sont extraits du document ainsi que "l'extrait de thème" du document (la ou les premières phrases du document). Ensuite, à partir de l'extrait de thème est extrait un ensemble unique de termes thématiques tels que les entités nommées, les phrases nominales et les mots uni-grammes. Ensuite, le plongement de mots/termes est utilisé pour obtenir la représentation vectorielle de l'extrait de thème qui est le résultat de l'addition des représentations vectorielles des termes thématiques. Le plongement de mots/termes est également utilisé pour obtenir la représentation vectorielle de chaque terme-clé candidat. Ensuite, un score, appelé poids thématique, est attribué à chaque candidat en utilisant la distance cosinus entre le vecteur de l'extrait de thème et le vecteur du candidat. Puis un graphe orienté est construit avec les termes-clés candidats comme nœuds et deux candidats sont reliés par une arête s'ils co-occurrent dans une fenêtre de taille 5. Les poids des arêtes sont calculés de la même manière que dans les travaux de Rui Wang *et al.* [Wang 2015]. Finalement, un algorithme PageRank pondéré est utilisé pour ordonner les termes-clés candidats et les N premiers sont considérés comme termes-clés. Contrairement à Rui Wang *et al.* [Wang 2015] qui ont utilisé des modèles de plongement de mots pré-entraînés (entraîné sur wikipédia par exemple), Mahata *et al.* [Mahata 2018] eux, ont entraîné leur propre plongement de termes sur des documents scientifiques.

Kamil Bennani-Smires *et al.* [Bennani-Smires 2018] proposent EmbedRank. Cette méthode est basée sur le plongement de termes/phrases. Après avoir extrait les termes-clés candidats à partir du document, ils les transforment en vecteur, puis ils représentent aussi le document en vecteur. Ensuite, le score de chaque candidat est calculé en utilisant une modification de la formule Maximal Marginal Relevance (MMR)

[Carbonell 1998] et les candidats ayant des scores élevés sont considérés comme termes-clés. La mesure de similarité entre un candidat et le document qu'ils ont utilisée dans MMR est basée sur la distance cosinus entre deux vecteurs. Ils ont aussi utilisé des modèles pré-entraînés de plongement de termes/phrases dans leurs travaux.

Sun *et al.* [Sun 2020] présentent une méthode, appelée SIFRank, basée sur un modèle de langue pré-entraîné. Plus précisément, elle combine le modèle de plongement de phrases SIF [Arora 2017] et le modèle de langue auto-régressif ELMo [Peters 2018]. SIFRank donne de meilleurs résultats sur les documents courts. Pour les documents longs, Sun *et al.* améliore SIFRank en tenant compte de toutes les positions des occurrences des termes-clés candidats dans le document.

2.2.1.3 Bilan

Les méthodes supervisées reformulent la tâche d'extraction de termes-clés soit en une tâche de classification, soit en une tâche d'ordonnancement. La tâche de classification consiste à classifier les termes-clés candidats en tant que "terme-clé" ou "non terme-clé", alors que la tâche d'ordonnancement consiste à attribuer un score aux termes-clés candidats puis à les ordonner et enfin considérer les N premiers en tant que termes-clés. Les méthodes supervisées utilisent des classifieurs et pour fonctionner, elles requièrent un corpus d'entraînement (une collection de documents) pour l'apprentissage. Ces méthodes donnent de bons résultats, mais dépendent fortement du volume de données d'entraînement ainsi que du domaine d'application.

Les méthodes non supervisées, quant à elles, reformulent la tâche d'extraction de termes-clés en une tâche d'ordonnancement. Elles ne requièrent pas de corpus d'entraînement donc elles sont plus généralisables (sauf celles basées sur le plongement de mots).

Dans ce travail de thèse, nous nous intéressons aux méthodes non supervisées d'extraction automatiques à base de graphes du fait de leurs bonnes performances malgré leur simplicité et la facilité de leur implantation. Ce sont aussi les plus courantes et elles sont suffisamment diversifiées. Cependant, les méthodes de la littérature présentent une, deux ou toutes des limites suivantes : (1) la construction du graphe de mots est basée sur la co-occurrence ; celle-ci ne capture pas très bien la relation sémantique entre deux mots dans un document car un document est souvent trop court pour le permettre (souvent un corpus de documents est nécessaire) ; (2) l'attribution des scores aux termes-clés candidats est fonction de leur longueur ; cela favorise souvent les candidats les plus longs ou les plus courts ; (3) le calcul des scores des termes-clés candidats est fonction des scores des mots qui les composent ; cela favorise la présence

de chevauchements des termes-clés puisque les candidats qui possèdent plusieurs mots en commun auront des scores proches ; et enfin (4) la construction du graphe de mots basée sur la co-occurrence entraîne aussi la perte d'informations qui sont peut-être importantes pour le choix des termes-clés telles que la fréquence ou la position des termes-clés candidats dans le document.

Rui Wang *et al.* [Wang 2015] résolvent le problème (1) en utilisant le plongement de mots lors de la construction du graphe de mots. Ensuite, ils utilisent un algorithme PageRank pondéré pour ordonner les nœuds du graphe et calculent le score d'un terme-clé candidat. Le terme-clé candidat est défini comme une séquence de noms et d'adjectifs se terminant uniquement par un nom. Son score est calculé en sommant les scores des mots qui le composent. Enfin, les termes-clés sont les premiers N candidats qui ont les scores les plus élevés.

Yeom *et al.* [Yeom 2019], quant à eux, résolvent les problèmes (2), (3) et (4) en modifiant la méthode SingleRank. En effet, pour résoudre le problème (2), ils adoptent une moyenne harmonique modifiée pour calculer les scores des termes-clés candidats au lieu de la somme qu'utilise SingleRank. Pour résoudre les problèmes (3) et (4), ils utilisent une modification de la méthode C-value pour recalculer les scores des termes-clés candidats obtenus par la moyenne harmonique.

Florescu *et al.* [Florescu 2017] résolvent aussi le problème (4) en modifiant simplement PageRank en tenant compte de toutes les positions des occurrences des mots dans le document.

Dans notre travail, nous proposons une méthode à base de graphes qui ne présente aucune des quatre limites citées précédemment. Pour cela, nous allons combiner les solutions apportées par Rui Wang *et al.* [Wang 2015] et Yeom *et al.* [Yeom 2019]. Cependant, nous proposons une façon d'intégrer le plongment de mots, autre que celles proposées par Rui Wang *et al.*. Aussi, nous utilisons les mesures de centralité pour ordonner les nœuds du graphe de mots, comme dans le travail de Boudin *et al.* [Boudin 2013a], au lieu de l'algorithme PageRank pondéré (voir section 2.3).

Dans ce qui suit, nous présentons d'abord le concept de plongement de mots avant de détailler notre méthode d'extraction de termes-clés.

2.2.2 Plongement de mots

Le Traitement Automatique des Langues (TAL) est un domaine multidisciplinaire alliant la linguistique et l'informatique. Dans plusieurs applications de TAL, la représentation des textes par des vecteurs est utilisée. Elle peut s'appliquer à différents niveaux : un corpus de documents, un document, une phrase, un mot.

L'une des représentations de textes la plus utilisée et la plus simple est celle dite par "**sac de mots**". Le sac de mots permet de représenter un texte par un vecteur dans l'espace des mots ou termes d'un dictionnaire construit au préalable ; les poids (coordonnées de ce vecteur) correspondent alors aux fréquences des mots dans le texte, éventuellement en suivant des calculs spécifiques comme la pondération Tf-Idf. Ainsi, par exemple, la i-ème composante du vecteur indique la fréquence du i-ème mot du dictionnaire dans le texte. Avec cette approche, la constitution du dictionnaire est une étape très importante qui a un impact sur les futures performances des systèmes qui vont utiliser la représentation. Bien que l'approche sac de mots soit très utilisée, cette approche présente aussi d'importantes lacunes dans la capture d'informations sur la structure des textes ainsi que les relations entre les mots qui les composent, les mots étant considérés comme étant indépendants.

De nombreuses méthodes ont été proposées pour représenter les mots par un vecteur dense afin de capturer les relations sémantiques entre eux. Ces représentations sont appelées les "**plongements de mots**" ou "word embedding". Le plongement de mots se base sur l'hypothèse que les mots qui sont utilisés dans les mêmes contextes tendent à avoir les mêmes significations. Les méthodes de construction des plongements de mots peuvent être catégorisées en deux types selon Baroni *et al.* [Baroni 2014] : basées sur des modèles de comptage comme le Latent Semantic Analysis (LSA), et basées sur des modèles de prédiction (la plupart utilisent les réseaux de neurones) comme Word2Vec [Mikolov 2013b]. La différence entre ces deux types d'approche se trouve au niveau de la construction des vecteurs de mots ainsi que sur le contexte pris en compte.

Les méthodes basées sur les modèles de comptage utilisent souvent tout le document comme contexte, voire tous les mots dans le corpus de documents. Elles présentent alors l'avantage de très bien prendre en compte des informations statistiques globales (par exemple avec l'utilisation du nombre total de co-occurrences dans le corpus). Par contre, ces approches font relativement mal la tâche d'analogie entre les mots alors que celles basées sur les modèles de prédiction le font très bien. Cependant, ces dernières ne prennent pas souvent en compte les informations statistiques globales parce qu'elles utilisent les mots voisins dans une fenêtre de mots comme contexte (par exemple, seul le nombre de co-occurrences dans la fenêtre de mots est considéré). Les approches basées sur les modèles de comptage tendent à être utilisées pour la modélisation de thèmes alors que les approches basées sur les modèles de prédiction sont plus efficaces pour obtenir la similarité entre les mots.

Dans la littérature, plusieurs méthodes ont été proposées pour construire les plongements de mots. Parmi elles, l'approche de Bengio *et al.* [Bengio 2003] consiste à apprendre à un réseau de neurones à estimer la probabilité du prochain mot, en s'ap-

puyant sur les mots qui le précèdent. Cette approche assigne donc une probabilité pour chaque mot dans le vocabulaire du corpus d'entraînement.

Une autre approche est celle de Schwenk [Schwenk 2007], nommée Continuous Space Language Models (CSLM). Elle modifie l'approche de Bengio *et al.* [Bengio 2003] en restreignant le calcul de la probabilité aux seuls mots les plus fréquents du vocabulaire. Collobert et Weston [Collobert 2008] proposent une autre solution pour éviter le calcul de la probabilité sur tout le vocabulaire à la couche de sortie. Pour cela, ils ont utilisé une autre fonction de coût pour l'apprentissage et leur approche estime la probabilité pour qu'un mot au milieu d'un N-gramme soit lié à son contexte. Turian *et al.* [Turian 2010] a revisité l'approche de Collobert et Weston [Collobert 2008] en proposant d'estimer la probabilité du dernier mot du N-gramme et en utilisant des taux d'apprentissage différents pour les poids du réseau de neurones.

Dans les sections suivantes, nous détaillons deux approches de plongement de mots qui sont parmi les plus utilisées en TAL et Recherche d'Information (RI) : Word2Vec [Mikolov 2013b] et Glove [Pennington 2014]. Ces deux approches de plongement de mots sont celles que nous allons utiliser dans la méthode que nous proposons dans ce travail.

2.2.2.1 Représentation Word2Vec

Word2Vec est une approche de représentation vectorielle de mots introduite par Mikolov *et al.* [Mikolov 2013a]. Cette approche s'appuie sur les réseaux de neurones et propose deux architectures : le modèle Countinious Bag of Words (CBOW) ou sac de mots continu et le modèle Skip-Gram. Ces modèles sont entraînés à partir des "mots centraux" et du contexte dans lequel ces derniers apparaissent, c'est-à-dire les mots qui les précèdent et qui les succèdent dans une fenêtre de C mots. Plus précisément, le modèle CBOW va prédire le mot central à partir de ses mots voisins (contexte) tandis que c'est l'inverse pour le modèle Skip-Gram qui va prédire les mots du contexte à partir du mot central.

Ces modèles sont structurés en trois couches : couche d'entrée, couche intermédiaire et la couche de sortie. La figure 2.2 présente la différence d'architecture entre les deux modèles.

Couche d'entrée : correspond à un sac de mots contenant les mots voisins du mot central pour le modèle CBOW et le mot central pour le modèle Skip-Gram.

Couche intermédiaire : correspond à la projection des mots d'entrée dans la matrice des poids.

Couche de sortie : correspond à la prédiction du modèle en utilisant la fonction Soft-

max. C'est-à-dire le sac de mots voisins pour le modèle Skip-Gram et le mot central pour le modèle CBOW.

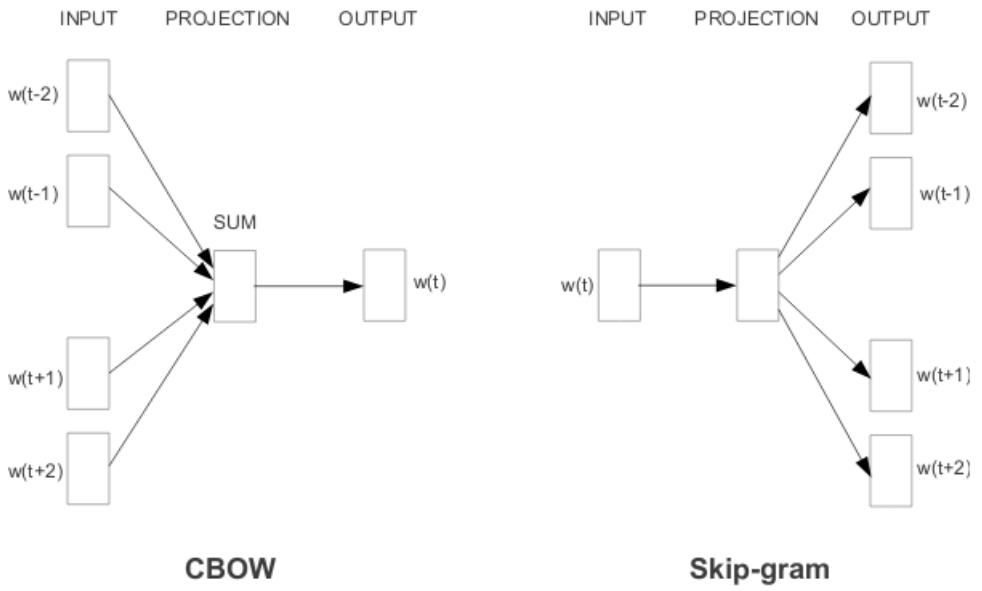


FIGURE 2.2 – Architecture des deux modèles, CBOW et Skip-Gram, Word2Vec [Mikolov 2013a]

L'idée générale est de définir un réseau de neurones qui a comme entrée X et comme sortie Y, qui sont des vecteurs de taille V (taille du vocabulaire) tel que chaque vecteur possède un unique 1. L'objectif est donc de calculer la matrice M de taille $V \times Q$ qui est connectée à gauche des couches de projections et à droite des couches d'entrée. Une ligne de cette matrice correspond à un vecteur d'un mot dans le vocabulaire. Q est un paramètre d'apprentissage, mais c'est également la dimension du vecteur, représentant un mot et obtenu par le modèle Word2Vec.

CBOW

L'architecture du modèle CBOW apprend à prédire un mot central w_t en fonction de son contexte $\{w_{t-2}, w_{t-1}, ..., w_{t+1}, w_{t+2}\}$, qui correspond aux mots précédents et aux mots suivants, tel que l'ensemble $\{w_{t-2}, w_{t-1}, ..., w_{t+1}, w_{t+2}\} \cup \{w_t\}$ est une phrase du langage ou une fenêtre de mots. Chaque mot w_i du contexte est représenté à la couche d'entrée par un vecteur $X_i : (x_1, x_2, ..., x_k, ..., x_V)$ où V est la taille du vocabulaire, k est le rang du mot dans le vocabulaire, $x_k = 1$ et $x_m = 0\, m \neq k$. Le passage à la couche de projection ou intermédiaire, qui est partagée par tous les mots, se fait par le produit

de la moyenne des vecteurs des mots du contexte et de la matrice de pondération entre la couche d'entrée et la couche de projection :

$$h = \frac{1}{C} \sum_{i=1}^{C} X_i \cdot M \tag{2.3}$$

où h est un vecteur de dimension Q correspondant à la projection du contexte sur la couche intermédiaire, C est la taille du contexte, c'est-à-dire le nombre de mots dans le contexte et M est matrice de taille $V \times Q$ correspondant à la matrice de passage de la couche d'entrée vers la couche intermédiaire.

Ensuite, la couche de sortie est calculée en utilisant la fonction "Softmax" de la manière suivante :

$$y_j = \frac{\exp^{(h \cdot M'_j)}}{\sum_{k=1}^{V} \exp^{(h \cdot M'_k)}} \tag{2.4}$$

où y_j est la $j^{ème}$ composante du vecteur Y de taille V de la couche de sortie et qui correspond à la probabilité que le $j^{ème}$ mot du vocabulaire appartienne au contexte de la couche d'entrée, c'est-à-dire la probabilité pour qu'il soit le mot central. M'_j est la $j^{ème}$ ligne de la matrice $M^{'}$ qui est la matrice, de taille $Q \times V$, de pondération entre la couche intermédiaire et la couche de sortie.

Par la suite, le modèle compare sa prédiction avec la réalité qui est un vecteur de taille V contenant un unique 1 à la $k^{ème}$ ligne tel que k est le rang du mot central dans le vocabulaire. Puis il corrige la représentation vectorielle du mot (sa prédiction) par rétro-propagation du gradient en ajustant les composantes des matrices de pondération M et $M^{'}$.

Ce modèle est nommé CBOW parce que d'une part, l'ordre des mots du contexte n'a pas d'influence sur la projection comme dans les modèles de sac de mots standards, et d'autre part, il utilise une représentation distribuée et continue du contexte, qui est différente des modèles de sac de mots standards.

Skip-Gram

L'architecture Skip-Gram est l'image inverse de l'architecture CBOW. Le mot central cible w_c est dorénavant utilisé comme entrée, et les mots du contexte sont à prédire en sortie. Dans cette architecture, la couche d'entrée est constituée d'un seul vecteur

$X : (x_1, x_2, ..., x_k, ..., x_V)$ représentant le mot central tel que $x_k = 1$ et $x_m = 0$ pour tout $m \neq k$. Le passage à la couche de projection se fait par le produit de l'entrée et de la matrice de pondération entre la couche d'entrée et la couche de projection :

$$h = X \times M \tag{2.5}$$

où M est matrice de taille $V \times Q$ correspondant à la matrice de passage de la couche d'entrée vers la couche intermédiaire.

La couche de sortie est aussi calculée en utilisant la fonction "Softmax" de la manière suivante :

$$y_{t,j} = \frac{\exp^{(h \cdot M'_j)}}{\sum_{k=1}^{V} \exp^{(h \cdot M'_k)}} \tag{2.6}$$

où $y_{t,j}$ est la $j^{ème}$ composante du vecteur représentant le $t^{ème}$ mot dans le contexte. M'_j est la $j^{ème}$ ligne de la matrice M' qui est la matrice de pondération entre la couche intermédiaire et la couche de sortie.

De la même manière que pour le modèle CBOW, les composantes des matrices de pondération M et M' sont aussi ajustées pour optimiser la capacité de prédiction du modèle en appliquant la rétro-propagation.

2.2.2.2 Représentation Glove

GloVe est un algorithme d'apprentissage non supervisé, proposé par Pennington *et al.* [Pennington 2014] pour la représentation vectorielle de mots. GloVe fait partie des méthodes qui combinent le modèle de comptage et le modèle de prédiction. L'idée est de construire un modèle qui exploite le principal avantage des modèles de comptage (les statistiques globales) tout en capturant simultanément les sous-structures linéaires importantes prévalant dans les modèles de prédiction. En effet, GloVe prend en compte toutes les informations portées par le corpus et pas seulement les informations portées par une fenêtre de mots.

Cette approche entraîne donc un modèle sur les statistiques de co-occurrences globales des mots provenant d'un corpus, c'est-à-dire en traitant le corpus en utilisant une fenêtre contextuelle glissante. Pour cela, une matrice M de co-occurrence globale des mots est construite où chaque élément M_{ij} correspond au nombre de fois où le mot

w_j apparaît dans le même contexte que le mot w_i. Une fois la matrice construite, un modèle de régression par la méthode des moindres carrés est entraîné pour construire des représentations vectorielles des mots w_i et w_j, notées respectivement $\overrightarrow{w_i}$ et $\overrightarrow{w_j}$, sachant que les informations sur la co-occurrence des mots w_j et w_i , doivent être conservées par ces représentations. Pour cela, Pennington *et al.* [Pennington 2014] proposent l'équation 2.7 :

$$\overrightarrow{w_i}^T \overrightarrow{w_j} + b_i + b_j = Log(M_{ij}) \tag{2.7}$$

où b_i (respectivement b_j) est le vecteur biais associé pour le mot w_i (respectivement w_j).

Cependant, l'équation 2.7 est mal définie puisque la fonction logarithme diverge quand son argument est égal à zéro. Même en utilisant la solution simple qui consiste à inclure un décalage additif dans le logarithme (ce qui permet d'éviter la divergence), l'équation 2.7 pondère toutes les co-occurrences de manière égale alors qu'elles n'ont pas la même qualité d'information. En effet, les co-occurrences très peu fréquentes ou rares ont tendance à être des bruits et portent moins d'informations que celles qui sont fréquentes. Il y aussi les mots, comme "tel" et "que", qui co-occurrent très fréquemment, mais qui ne portent pas d'informations importantes. Pour pallier à cela, Pennington *et al.* [Pennington 2014] proposent donc un nouveau modèle pondéré de régression par la méthode des moindres carrés qui va minimiser la fonction de coût de l'équation 2.8.

$$J = \sum_{i,j=1}^{V} f(M_{ij})(\overrightarrow{w_i}^T \overrightarrow{w_j} + b_i + b_j = Log(M_{ij}))^2 \tag{2.8}$$

où V est la taille du vocabulaire et f une fonction de pondération définie dans l'équation 2.9.

$$f(M_{ij}) = \begin{cases} \left(\frac{M_{ij}}{M_{max}}\right)^{\alpha} & si\ M_{ij} < M_{max} \\ 1 & sinon \end{cases} \tag{2.9}$$

où $M_{max} = 100$ et $\alpha = \frac{3}{4}$

Cette fonction de pondération permet donc de donner un poids important aux paires de mots qui co-occurrent fréquemment, mais limite aussi le poids des paires qui co-occurrent trop souvent. Pour cela, la fonction retourne soit un poids (entre 0 et 1), qui est calculé en tenant compte de la valeur de co-occurrence M_{ij} et où la distribution de poids dans cette plage est décidée par α, soit simplement 1 si cette valeur de co-occurrence est supérieure à une valeur maximale M_{max}.

2.3 Contributions

Dans cette section, nous présentons une méthode non supervisée à base de graphes pour l'extraction de termes-clés qui ne présente aucune des limites des méthodes à base de graphes de la littérature à savoir : les méthodes de la littérature

(1) sont généralement basées sur la co-occurrence des unités textuelles dans une fenêtre d'une certaine taille. Le problème est qu'une unité textuelle qui a une grande fréquence dans le document aura un score élevé par rapport aux autres, même si elle n'a pas une grande qualité de recouvrement et de représentation du document.

(2) attribuent des scores aux termes-clés candidats en fonction de leur longueur. En effet, le score d'un candidat est la somme ou la moyenne des scores des mots qu'il contient. Cela est problématique car favorisant souvent les candidats les plus longs ou les plus courts.

(3) produisent des termes-clés redondants alors ils doivent être complémentaires. Ceci est causé par le chevauchement de certains termes-clés candidats c'est-à-dire que certains candidats sont des sous-chaînes d'autres candidats.

(4) perdent des informations sur les termes-clés candidats, telles que la fréquence ou la position des termes-clés candidats, lorsque le document est représenté sous forme de graphe, et ces informations peuvent être importantes. Par exemple, Florescu *et. al* [Florescu 2017] ont émis l'hypothèse que les termes-clés apparaissent fréquemment à des positions très proches du début du document, voire dans le titre.

Dans la littérature, ces problèmes ont été déjà résolus mais séparément : Rui Wang *et al.* [Wang 2015] propose d'intégrer le plongement de mots pour résoudre le problème (1) ; Yeom *et al.* [Yeom 2019] proposent d'utiliser une moyenne harmonique pour résoudre le problème (2) et la méthode C-value pour résoudre les problèmes (3) et (4). Notre méthode, quant à elle, combine les solutions apportées par Rui Wang *et al.* [Wang 2015] et Yeom *et al.* [Yeom 2019]. Cependant la façon d'intégrer le plongment de mots diffère de celles proposées par Rui Wang *et al.* ainsi que la méthode d'ordon-

nancement des nœuds.

Dans ce qui suit, nous allons détailler toutes les étapes de la méthode d'extraction de termes-clés à base de graphes que nous avons adoptées. Nous avons retenu le découpage en étapes de [Boudin 2013a] : prétraitements, construction du graphe de mots, ordonnancement des nœuds du graphe, construction des termes-clés candidats, ordonnancement des termes-clés candidats et enfin la sélection des termes-clés.

2.3.1 Prétraitements

Quand on parle de prétraitements dans les traitements ou analyse de textes, on parle souvent de suppression des mots vides et des ponctuations, de la racinisation ou dé-suffixation, de lemmatisation, etc. Ces prétraitements sont aussi utilisés pour l'extraction automatique de termes-clés, mais l'un des prétraitements le plus utilisé, et parfois crucial, est l'étiquetage morpho-syntaxique des mots. Ces étiquettes sont très importantes durant l'extraction des termes-clés candidats ou la construction du graphe de mots pour les méthodes à base de graphes. Durant notre travail, afin d'étiqueter tous les mots dans le document, nous avons utilisé Stanford CoreNLP suite v3.6.0.

Nous avons aussi effectué une racinisation en utilisant la méthode de Porter implémentée dans la librairie nltk de python et aussi une lemmatisation en utilisant le wordnet lemmatizer, qui utilise la base de données WordNet pour rechercher des lemmes de mots, fourni par la librairie nltk de python. Nous avons effectué ces prétraitements afin de voir s'ils auront un effet sur les résultats. Dans l'évaluation, nous comparerons donc les résultats des méthodes d'extraction des termes-clés quand on réalise : (a) une racinisation, (b) une lemmatisation, et (c) aucun des deux prétraitements.

2.3.2 Construction du graphe de mots

Un graphe de mots est construit à partir des mots résultant de la première étape. Nous ne gardons que les mots qui sont étiquetés comme nom ou adjectif pour construire le graphe. En effet, chaque nœud du graphe représente un mot et deux nœuds sont reliés entre eux si les mots qu'ils représentent co-occurrent dans une fenêtre de mots dans le document.

Par exemple pour un document contenant le texte : "Une légère brise de côte pourra faiblement rafraîchir une atmosphère terrestre", deux graphes de mots sont présentés dans la figure 2.3 : celui de gauche pour une fenêtre de co-occurrence de 3 mots et celui de droite pour une fenêtre de 7 mots.

Nous avons étudié trois méthodes de pondération durant nos expérimentations

(a) Fenêtre de 3 mots. (b) Fenêtre de 7 mots.

FIGURE 2.3 – Exemple de graphes de mots

dont une est la méthode par défaut, c'est-à-dire la co-occurrence, et les deux autres sont notre contribution.

(i) **Co-occurrence** : le poids de l'arête est le nombre de co-occurrences des deux mots (nœuds) dans une fenêtre de co-occurrence de mots comme dans les travaux de Boudin *et al.* [Boudin 2013a].

(ii) **Co-occurrence avec le plongement de mots** : le poids de l'arête entre deux nœuds est obtenu par le produit du nombre de co-occurrences des deux mots par la similarité cosinus des deux vecteurs représentant les deux mots (nœuds). L'idée derrière cette approche est de renforcer le lien sémantique entre deux mots par le nombre de fois qu'ils co-occurrent dans le document.

(iii) **Le plongement de mots seulement** : le poids de l'arête entre deux nœuds est la valeur de la similarité cosinus des deux vecteurs représentant les deux mots (nœuds). L'intuition derrière cette approche est de s'appuyer entièrement sur le plongement de mots pour quantifier la relation entre deux nœuds.

Notre choix s'est arrêté sur la mesure de similarité cosinus du fait de sa simplicité et son efficacité dans les autres domaines du TAL comme la traduction automatique et l'analyse de sentiments. Dans leur travail aussi, Rui Wang *et al.* [Wang 2015] ont rapporté que c'est la mesure de similarité cosinus qui donne de meilleurs résultats par rapport à la mesure de similarité euclidienne.

La mesure de similarité cosinus est calculée avec l'équation suivante :

$$cos = \frac{v_i \cdot v_j}{\|v_i\| \cdot \|v_j\|} \tag{2.10}$$

où v_i (respectivement v_j) est la représentation vectorielle, obtenue en utilisant le plon-

gement de mots, du mot w_i (respectivement w_j).

2.3.3 Ordonnancement des nœuds

Une fois le graphe de mots construit dans l'étape précédente, il faut maintenant attribuer à chaque nœud un score d'importance en appliquant un algorithme d'ordonnancement au graphe.

Plusieurs algorithmes d'ordonnancement existent, mais durant nos expérimentations nous avons utilisé et comparé les mesures de centralité qui sont aussi utilisées dans le travail de Boudin *et. al* [Boudin 2013a]. En effet, ces méthodes sont simples mais efficaces puisqu'ils obtiennent des résultats comparables à l'algorithme largement utilisé TextRank.

Nous allons détailler ces mesures de centralité et pour cela, nous allons noter le graphe de mots $G = (V, E)$ où $V = \{V_1, V_2, ..., V_i\}$ l'ensemble des nœuds et $|V|$ est le nombre de nœuds.

- *Centralité de proximité* : est définie comme l'inverse de la somme des plus courtes distances entre un nœud et tous les autres nœuds du graphe [Bavelas 1950]. Elle mesure la proximité d'un nœud par rapport aux autres nœuds dans le graphe, ainsi un nœud avec une mesure de centralité de proximité élevée possède les chemins les plus courts vers les autres nœuds. Elle est donnée par la formule :

$$C_c(V_i) = \frac{|V| - 1}{\sum_{V_j \neq V_i} d(V_i, V_j)} \tag{2.11}$$

  où $d(V_i, V_j)$ est le plus court chemin entre les nœds V_i et V_j.
- *Centralité de vecteur propre* : mesure le niveau d'influence d'un nœud dans le graphe. Elle attribue des scores relatifs à tous les nœuds selon le concept suivant : être connecté aux nœuds ayant des scores élevés permet, au nœud en question, d'avoir un meilleur score que d'être connecté à des nœuds ayant des faibles scores [Bonacich 1987]. Cette centralité est donnée par la formule :

$$C_{ev}(V_i) = \frac{1}{\lambda} \sum_{V_j \epsilon M(V_i)} p_{i,j} \times C_{ev} d(V_j) \tag{2.12}$$

  où $M(V_i)$ est l'ensemble des voisins du nœud V_i, $p_{i,j}$ est le poids de l'arrête reliant les nœuds V_i et V_j, et λ est une constante.

- *Centralité d'intermédiarité* : quantifie le nombre de fois qu'un nœud est considéré comme un pont ou point de passage suivant le plus court chemin entre deux autres nœuds [Freeman 1977]. Elle est donnée par la formule :

$$C_b(V_i) = 2 \times \frac{\sum_{V_j \neq V_i \neq V_k \epsilon V} \frac{\sigma_{jk}(V_i)}{\sigma_{jk}}}{(|V|-1)(|V|-2)} \tag{2.13}$$

où σ_{jk} est le nombre total de plus courts chemins reliant les nœuds V_j et V_k et $\sigma_{jk}(V_i)$ est le nombre de ces courts chemins qui passent par le nœud V_i.

- *Hits* : mesure l'importance d'un nœud en utilisant deux scores : *score de hub* (somme des scores d'autorité des nœuds sur lesquels il pointe) et *score d'autorité* (somme des scores de hub des nœuds qui le pointent) [Kleinberg 1999]. Dans notre travail, nous considérons le score d'autorité qui est donné par la formule :

$$Aut(V_i) = \sum_{V_j \epsilon M(V_i)} = Connect(V_i) \tag{2.14}$$

tel que

$$Connect(V_i) = \sum_{V_j \epsilon M(V_i)} = Aut(V_i) \tag{2.15}$$

où $Aut(V_i)$ est le score d'autorité du nœud V_i, $Connect(V_i)$ le score de hub, et $M(V_i)$ est l'ensemble des voisins du nœud V_i.

- *Centralité degré* : est la mesure de centralité la plus simple conceptuellement. Elle est définie pour un nœud donné comme le nombre de ses liens incidents, c'est-à-dire le nombre de ses voisins.

Après cette étape, à chaque mot/nœud est associé un score ; ces scores seront utilisés lors de l'ordonnancement des termes-clés candidats.

2.3.4 Sélection ou construction des termes-clés candidats

La sélection des termes-clés candidats consiste à établir une liste de mots ou de groupes de mots possibles pour un document donné. Les termes-clés sont ensuite sélectionnés ou générés à partir de cette liste. Cette étape est critique parce qu'un nombre insuffisant de termes-clés candidats sélectionnés diminue la performance de l'extraction de termes-clés. Inversement, un nombre trop important de termes-clés candidats

augmente la difficulté de l'extraction [Hasan 2014].

Plusieurs méthodes existent pour la sélection de termes-clés candidats et de nombreux travaux ont montré que les groupes nominaux approximés par des séquences d'adjectifs et de noms permettent d'obtenir de bons termes-clés candidats très proches des termes-clés de référence.

Ainsi, dans notre travail, nous avons choisi trois méthodes de sélection de termes-clés candidats :

(a) la plus longue séquence d'adjectifs et de noms : toutes les phrases du document sont parcourues mot par mot et on arrête dès qu'un mot qui n'est pas un adjectif ou un nom apparaît. Alors la séquence de mots précédente, si elle existe, devient un terme-clé candidat. Puis, le parcours reprend en commençant par le mot suivant et ainsi de suite jusqu'à la fin du document.

(b) le patron grammatical "<ADJ>*<NOUN|PROPN>+" : ce patron permet d'extraire les séquences de noms ou de pronoms modifiées de façon optionnelle par un ou plusieurs adjectifs antéposés.

(c) sélection des termes avec le même patron grammatical que précédemment ("<ADJ>*<NOUN|PROPN>+") puis ceux qui sont composés de plus de 3 mots ne sont pas retenus comme termes-clés candidats.

Il faut noter que dans notre cas les termes-clés candidats sont strictement composés des mots représentant les nœuds du graphe de mots que nous avons construit dans l'étape précédente.

2.3.5 Ordonnancement des termes-clés candidats

Les termes-clés candidats formés, pour calculer leur score, il suffit d'utiliser les scores des mots obtenus avec l'étape d'ordonnancement des nœuds. Dans la littérature, le score d'un terme-clé candidat est la somme ou la moyenne des scores des mots qui le composent. Cependant, cette technique favorise les candidats les plus longs ou les plus courts. Pour pallier ceci, nous adoptons pour la méthode proposée par Yeom *et al.* [Yeom 2019] qui est l'utilisation d'une moyenne harmonique modifiée au lieu de la somme ou de la moyenne. Le calcul du score d'un terme-clé candidat est donc réalisé avec la formule suivante :

$$ScoreCand(candidat) = \frac{|candidat|}{\sum_{w_i \in candidat} \frac{1}{scoreMot(w_i)}} * L(candidat) \tag{2.16}$$

Où

$$L(candidat) = \begin{cases} log_2 |candidat|, & si\ |candidat| > 1 \\ \beta,\ où\ 0 < \beta < 1, & si\ |candidat| = 1 \end{cases} \tag{2.17}$$

et $candidat$ désigne un terme-clé candidat composé de $|candidat|$ mots : $\{w_1; w_2; \cdots; w_{|candidat|}\}$, $|candidat|$ représente le nombre de mots qui composent le terme-clé candidat, et $scoreMot(w_i)$ est le score du mot w_i.

Maintenant chaque terme-clé candidat possède un score et ils peuvent être ordonnés. Cependant, il y a encore le problème de chevauchements des candidats, c'est-à-dire que certains candidats sont des sous-chaînes d'autres candidats. Par exemple, les deux candidats suivants se chevauchent : "légère brise" et "légère brise de côte" et ils peuvent avoir des scores similaires et peuvent donc être considérés tous les deux comme des termes-clés ou bien le plus long candidat peut être avantagé. Il y a aussi le problème de pertes d'informations telles que la fréquence ou la position des termes-clés candidats.

Pour résoudre ces deux problèmes, nous utilisons la méthode que Yeom *et al.* [Yeom 2019] qui est de recalculer le score d'un terme-clé candidat avec la méthode modifiée C-value. Le score final d'un candidat est donc calculé avec la formule suivante :

$$FinalScore(candidat) = \begin{cases} L(candidat) * Score(candidat) & où\ candidat\ n'est\ pas\ une\ sous - chaîne \\ L(candidat) * (Score(candidat) - R) & où\ candidat\ est\ une\ sous - chaîne \end{cases} \tag{2.18}$$

où

$$R = \frac{1}{|T_{candidat}|} * \sum_{b_i \in T_{candidat}} Score(b_i) \tag{2.19}$$

et

$$Score(candidat) = ScoreCand(candidat) * freq(candidat) \tag{2.20}$$

et $candidat$ désigne un terme-clé candidat, $T_{candidat}$ est l'ensemble de termes-clés candidats qui contiennent $candidat$, b_i est un terme-clé candidat appartenant à $T_{candidat}$ (c'est-à-dire qui contient $candidat$), $|T_{candidat}|$ est le nombre de termes-clés candidats dans $T_{candidat}$, $freq(candidat)$ est la fréquence du terme-clé candidat dans le document, $L(candidat)$ est calculé avec l'équation (2.17) et $ScoreCand(candidat)$ est calculé avec l'équation (2.16).

Le chevauchement de deux termes-clés candidats est résolu dans l'équation 2.18

avec le signe − devant le R puisque cela enlève l'effet qu'un candidat soit une sous-chaîne d'un autre candidat plus long. Ensuite, le problème de perte d'information est résolu en intégrant la fréquence dans le score du candidat dans l'équation 2.20.

2.3.6 Sélection des termes-clés

Comme tous les termes-clés candidats ont chacun un score (le score final), ils sont ordonnés à l'aide de leurs scores (ordre décroissant). Les N premiers de ces candidats sont alors considérés comme les termes-clés. En d'autres termes, les termes-clés sont les candidats les plus importants, c'est-à-dire ceux qui ont les meilleurs scores.

2.4 Expérimentations et résultats

Dans cette section, nous allons détailler les expérimentations que nous avons réalisées et les résultats que nous avons obtenus. Nous présenterons aussi les collections de données utilisées ainsi que les mesures d'évaluation.

2.4.1 Collections de données

Les collections de données, composées de documents, sont indispensables à l'évaluation et à la comparaison par rapport aux méthodes de la littérature. En général, elles sont divisées en deux ensembles : un ensemble d'entraînement et un ensemble de test. L'ensemble d'entraînement est utilisé par les méthodes supervisées pour entraîner un modèle et peut être aussi utilisé pour configurer les paramètres des méthodes non supervisées. L'ensemble de test, qui contient aussi les termes-clés de référence, sert à évaluer les performances des méthodes.

Plusieurs et diverses collections sont accessibles publiquement. Cette diversité est très importante d'après Hasan *et. el.* [Hasan 2010] pour la compréhension des performances des méthodes d'extraction de termes-clés, c'est-à-dire comprendre les points faibles et les points forts de chaque méthode. D'après Hasan *et. el.* [Hasan 2014] encore, les performances peuvent être influencées par différents facteurs qui sont :

- la longueur du document : un document long possède un nombre élevé de termes-clés candidats qui donne la possibilité d'extraire beaucoup de termes-clés et inversement pour les documents courts ;
- la structure du document : une méthode qui tient compte de la structure sera avantagée sur les documents structurés ;

— le changement de thématiques : le changement de thématique dans un même document désavantagerait les méthodes qui utilisent et se basent sur la position de la première occurrence des termes-clés candidats ;
— la corrélation de thèmes : les méthodes qui se basent sur les liens sémantiques entre les termes-clés candidats seront désavantagées sur un document qui aborde des thèmes sans relation.

Bougouin *et al.* [Bougouin 2014] ont trouvé deux autres facteurs qui concernent cette fois-ci les termes-clés de référence :

— la présence dans le document : les méthodes d'extraction de termes-clés sont désavantagées dans le cas où la majorité des termes-clés de référence n'apparaissent pas dans le document ;
— la qualité : les différents types d'annotateurs changent les termes-clés de référence.

Dans nos travaux, nous utilisons donc onze collections différentes de données pour bien évaluer nos méthodes : NUS [Nguyen 2007], PubMed [Schutz 2008], ACM [Krapivin 2009], Citeulike-180 [Medelyan 2009], Semeval [Kim 2010], Inspec [Hulth 2003], TALN-Archives [Boudin 2013b], KDD [Caragea 2014], WWW [Caragea 2014], DUC-2001 [Wan 2008] et 500N-KPCrowd [Marujo 2012].

Le tableau 2.1 résume les caractéristiques des corpus que nous utiliserons dans nos travaux. Nous pouvons y voir la nature du corpus, c'est-à-dire le type de document dans le corpus. Il y a aussi le nombre de documents, la façon d'extraire les termes-clés associés aux documents (A : par les auteurs du document, L : par des lecteurs, A+L : par les auteurs et des lecteurs, et I : termes-clés non contrôlés de Inspec), le nombre moyen de termes-clés assignés par document, et enfin la taille (nombre de mots) moyenne d'un document dans le sous-ensemble de test du corpus.

La majorité des collections sont constituées d'articles scientifiques (collections NUS, PubMed, ACM, Citeulike-180 et Semeval) ou de résumés d'articles scientifiques (collections Inspec, TALN-Archives, KDD et WWW). Seules les collections DUC-2001 et 500N-KPCrowd sont des articles de presse. Par ailleurs, la majorité des collections sont constituées d'un seul ensemble ; seules les collections Semeval, Inspec et 500N-KPCrowd sont scindées en sous-collection d'apprentissage/validation et de test. Les évaluations de notre méthode ont été faites sur les collections de test. Comme indiqué dans le tableau 2.1, le nombre de documents des collections varie de 50 à 2304. Le nombre moyen de termes-clés de référence varie quant à lui de 4 à 46.2. Notons que la collection Semeval est issue de la campagne d'évaluation Semeval utilisée pour la tâche 5 de l'édition 2010 qui est dédiée à l'extraction automatique de termes-clés.

Durant l'évaluation (section 2.4.3), nous allons sélectionner les premiers N termes-

clés candidats comme termes-clés. Pour chaque corpus, la valeur de N que nous allons considérer est proche du nombre moyen de termes-clés assignés par document. C'est-à-dire que N vaut 5 pour les collections PubMed, ACM, Citeulike-180, TALN-Archives, KDD et WWW ; 10 pour les collections NUS, Inspec et DUC-2001 ; 15 pour le corpus Semeval et 45 pour la collection 500N-KPCrowd.

TABLEAU 2.1 – Caractéristiques des corpus.

Corpus	Nature	#documents	Annotation	#termes-clés (test)	#mots (test)
NUS [Nguyen 2007]	Papier complet	211	A+L	11.0	8398.3
PubMed [Schutz 2008]	Papier complet	1320	A	5.4	5322.9
ACM [Krapivin 2009]	Papier complet	2304	A	5.3	9197.6
Citeulike-180 [Medelyan 2009]	Papier complet	182	L	5.4	8589.7
Semeval [Kim 2010]	Papier complet	244	A+L	14.7	7961.2
Inspec [Hulth 2003]	Résumé	2000	I	9.8	134.6
TALN-Archives [Boudin 2013b]	Résumé	521	A	4.0	123.1
KDD [Caragea 2014]	Résumé	755	A	4.1	190.7
WWW [Caragea 2014]	Résumé	1330	A	4.8	163.5
DUC-2001 [Wan 2008]	Article de presse	308	L	8.1	847.2
500N-KPCrowd [Marujo 2012]	Article de presse	500	L	46.2	465.3

2.4.2 Mesures d'évaluation

Nous allons mesurer les performances des méthodes d'extraction automatique de termes-clés avec les mesures de précision (P), rappel (R) et f1-mesure (F). Ce sont aussi les mesures utilisées dans le cadre de la tâche Semeval [Kim 2010]. Pour calculer la performance d'une méthode, c'est-à-dire la précision (respectivement rappel), nous avons d'abord calculé la précision (respectivement rappel) pour chaque document, et puis la moyenne de ces valeurs. Pour la f1-mesure, il y a deux manières de la calculer : soit en calculant la f1-mesure pour chaque document, puis en effectuant la moyenne, soit en la calculant à l'aide de la moyenne du rappel et la moyenne de la précision. Dans notre travail, nous avons utilisé la première, c'est-à-dire la moyenne des f1-mesures des documents.

Ces mesures sont définies comme suit :

- **Précision (P)** : est définie par le nombre de termes-clés pertinents retrouvés par rapport au nombre total de termes-clés extraits.

$$P = \frac{termes - clés\ pertinents\ extraits}{termes - clés\ extraits} \tag{2.21}$$

- **Rappel (R)** : est défini par le nombre de termes-clés pertinents retrouvés par rapport au nombre total de termes-clés de référence du document.

$$R = \frac{termes - clés\ pertinents\ extraits}{termes - clés\ de\ référence} \tag{2.22}$$

- **f1-mesure (F)** : est la moyenne harmonique du rappel et de la précision, pondérés de façon égale.

$$F = 2 \times \frac{R \times P}{R + P} \tag{2.23}$$

2.4.3 Évaluation

Dans notre travail, nous réalisons deux types d'expériences : une pour déterminer les paramètres de nos approches et une seconde pour les comparer aux travaux de la littérature. Pour les jeux de données qui ne possèdent pas des ensembles d'entraînement, nous allons utiliser les paramètres déterminés sur un jeu de données similaires. Ainsi, pour les jeux de données contenant des papiers longs, les paramètres seront déterminés sur la collection Semeval ; pour ceux qui contiennent des résumés, ce sera sur la collection Inspec ; et enfin, pour ceux qui contiennent des articles de presse, ce sera sur la collection 500N-KPCrowd. Ainsi, nous transférons les paramètres d'une collection à une autre.

Rappelons que dans la suite dans ce document, nous allons extraire les premiers N termes-clés candidats comme termes-clés où N vaut 5 pour les collections PubMed, ACM, Citeulike-180, TALN-Archives, KDD et WWW ; 10 pour les collections NUS, Inspec et DUC-2001 ; 15 pour le corpus Semeval et 45 pour la collection 500N-KPCrowd.

Voici les paramètres à déterminer dans notre méthode :

- Le type prétraitement de textes : racinisation, lemmatisation ou aucun prétraitement des mots.
- La fenêtre de co-occurrence C de mots lors de la construction du graphe de mots. Nous avons fait varier C entre 2 et 10 avec un pas de 1.
- Le poids d'une arête dans le graphe qui est à choisir parmi les trois suivants :

(i) Co-occurrence, (ii) co-occurrence + plongement de mots, (iii) plongement de mots seulement.
- Le modèle pré-entraîné de plongement de mots à utiliser. Nous avons testé cinq modèles dont quatre modèles Glove (Glove_50, Glove_100, Glove_200, Glove_300) [1] et un modèle Word2Vec qui est celui entraîné par Google.
- La méthode de sélection de termes-clés candidats qui est à choisir parmi les trois suivantes : (a) longue séquence d'adjectifs et de noms, (b) patron grammatical, et (c) patron grammatical dont la longueur maximale d'un candidat est de trois mots.
- La valeur de β dans l'équation 2.16 (dans la formule $L(candidat)$ de l'équation 2.17) lors du calcul du score d'un terme-clé candidat avec la formule modifiée de la moyenne harmonique. Nous allons noter ce paramètre β_1 et le faisons varier entre 0.1 et 0.9 avec un pas de 0.1.
- La valeur de β dans l'équation 2.18 (dans la formule $L(candidat)$ de l'équation 2.17) lors du calcul du score final d'un terme-clé candidat. Nous allons noter ce paramètre β_2 et le faisons varier entre 0.1 et 0.9 avec un pas de 0.1.

Dans le tableau 2.2, nous pouvons voir les valeurs de chaque paramètre pour chaque jeu de données qui possède un ensemble d'entraînement. Les deux paramètres : *poids de l'arête* et *modèle de plongement de mots* sont dépendants. En effet, quand la méthode (i) Co-occurrence est utilisée pour pondérer les arêtes du graphe, il n'est pas nécessaire de trouver le paramètre *modèle de plongement de mots*.

De plus, avec la mesure de centralité degré, il n'est pas nécessaire de trouver les deux paramètres : *modèle de plongement de mots* et *poids de l'arête*, puisque cette mesure de centralité est seulement basée sur le nombre d'arêtes. Nous constatons dans ce tableau que le prétraitement avec la racinisation offre les meilleurs résultats quel que soit le jeu de données. Nous constatons aussi que pour le corpus Inspec, un terme-clé est une longue séquence d'adjectifs et de noms alors que pour les deux autres corpus, il suit un patron grammatical dont la longueur maximale est de trois mots. Aussi, dès que le plongement de mot est intégré dans la méthode, c'est souvent le modèle pré-entraîné Word2Vec qui est utilisé.

1. Glove_50 signifie que le modèle pré-entraîné de Glove fournit un vecteur de taille 50.

TABLEAU 2.2 – Choix des paramètres sur les jeux de données possédant un ensemble d'entraînement.

Méthodes	Paramètres	INSPEC	Semeval	500N-KPCrowd
Proximité	Prétraitement	racinisation	racinisation	racinisation
	Fenêtre C	6	7	8
	Plongement de mots	Word2Vec	Word2Vec	Glove_300
	Poids de l'arrête	(iii)	(iii)	(ii)
	Sélection candidats	(a)	(a)	(c)
	β_1	0.5	0.4	0.9
	β_2	0.1	0.3	0.9
Degré	Prétraitement	racinisation	racinisation	racinisation
	Fenêtre C	4	9	8
	Sélection candidats	(a)	(c)	(c)
	β_1	0.1	0.1	0.9
	β_2	0.5	0.8	0.9
Vecteur propre	Prétraitement	racinisation	racinisation	racinisation
	Fenêtre C	9	10	10
	Plongement de mots	-	Word2Vec	Word2Vec
	Poids de l'arrête	(i)	(iii)	(ii)
	Sélection candidats	(a)	(c)	(c)
	β_1	0.1	0.9	0.9
	β_2	0.1	0.1	0.9
Intermédiarité	Prétraitement	racinisation	racinisation	racinisation
	Fenêtre C	3	2	9
	Plongement de mots	Word2Vec	-	-
	Poids de l'arrête	(iii)	(i)	(i)
	Sélection candidats	(a)	(c)	(c)
	β_1	0.1	0.6	0.9
	β_2	0.3	0.3	0.9

TABLEAU 2.2 – suite de la page précédente

Méthodes	Paramètres	INSPEC	Semeval	500N-KPCrowd
Hits	Prétraitement	racinisation	racinisation	racinisation
	Fenêtre C	9	6	8
	Plongement de mots	Word2Vec	Word2Vec	-
	Poids de l'arrête	(iii)	(iii)	(i)
	Sélection candidats	(a)	(c)	(c)
	β_1	0.2	0.2	0.9
	β_2	0.1	0.7	0.9

Le tableau 2.3 reporte la comparaison des résultats de notre approche par rapport à la méthode de référence, celle de Boudin *et al.* [Boudin 2013a], sur les 11 jeux de données. * indique une différence statistiquement significative avec le test de Student apparié avec la valeur p plus petite que 0.05.

Comme on peut voir dans le tableau 2.3, notre approche surpasse les résultats de Boudin *et al.* [Boudin 2013a] sur tous les jeux de données contenant des documents longs, sur un jeu de données contenant des documents courts (Inspec) et sur un jeu de données contenant des documents de type article de presse (500N-KPCrowd). Toutes ces améliorations sont statistiquement significatives sauf sur 500N-KPCrowd où seule notre approche avec la mesure de centralité "proximité" fournit une amélioration significative. Sur les autres jeux de données, 3 contenant des documents courts et un contenant des articles de presse, notre approche est surpassée, et cela significativement.

Au final, nous pouvons voir sur ces résultats que les modifications que nous avons apportées à l'approche de Boudin *et al.* [Boudin 2013a] fonctionnent mieux sur les documents longs que sur les documents courts. Cependant, nous constatons aussi que sur Inspec, notre approche fonctionne aussi très bien. C'est peut-être lié au fait que la recherche de paramètres a été réalisée sur les données d'entraînement de cette collection.

TABLEAU 2.3 – Comparaison par rapport aux résultats de [Boudin 2013a] qui utilise les mêmes mesures de centralité que notre méthode. Les meilleurs résultats sont en gras. * indique une différence statistiquement significative avec le test de Student apparié (valeur p < 0.05).

	Méthodes	Notre			[Boudin 2013a]		
		P	R	F	P	R	F
Inspec	Proximité	0.361	0.421	**0.389***	0.338	0.393	0.363
	Degré	0.359	0.418	**0.386***	0.311	0.370	0.338
	Vecteur propre	0.354	0.412	**0.381***	0.302	0.359	0.328
	Intermédiarité	0.341	0.394	**0.366***	0.282	0.337	0.307
	Hits	0.357	0.413	**0.383***	0.302	0.359	0.328
TALN-Archives	Proximité	0.098	0.132	0.112	0.127	0.169	**0.145***
	Degré	0.113	0.152	0.129	0.124	0.163	**0.141***
	Vecteur propre	0.122	0.164	0.14	0.136	0.177	**0.154**
	Intermédiarité	0.102	0.136	0.117	0.12	0.156	**0.135***
	Hits	0.108	0.147	0.125	0.136	0.177	**0.153***
KDD	Proximité	0.073	0.095	0.082	0.091	0.117	**0.102***
	Degré	0.09	0.116	0.101	0.097	0.124	**0.109**
	Vecteur propre	0.095	0.124	0.108	0.104	0.133	**0.116***
	Intermédiarité	0.063	0.081	0.071	0.094	0.118	**0.105***
	Hits	0.083	0.108	0.094	0.103	0.133	**0.116***
WWW	Proximité	0.073	0.087	0.079	0.091	0.106	**0.098***
	Degré	0.085	0.101	0.093	0.107	0.122	**0.114***
	Vecteur propre	0.094	0.110	0.101	0.115	0.130	**0.122***
	Intermédiarité	0.064	0.074	0.069	0.105	0.116	**0.111***
	Hits	0.078	0.093	0.085	0.115	0.130	**0.122***
500N-KPCrowd	Proximité	0.240	0.256	**0.247***	0.216	0.242	0.228
	Degré	0.238	0.258	**0.247**	0.234	0.259	0.246
	Vecteur propre	0.248	0.266	**0.257**	0.241	0.262	0.251
	Intermédiarité	0.245	0.264	**0.254**	0.240	0.264	0.252
	Hits	0.243	0.262	**0.252**	0.242	0.263	0.252

TABLEAU 2.3 – suite de la page précédente

	Méthodes	Notre			[Boudin 2013a]		
		P	R	F	P	R	F
DUC-2001	Proximité	0.140	0.186	0.160	0.226	0.288	**0.253***
	Degré	0.167	0.222	0.190	0.240	0.308	**0.269***
	Vecteur propre	0.174	0.229	0.198	0.221	0.284	**0.249***
	Intermédiarité	0.156	0.205	0.177	0.226	0.293	**0.255***
	Hits	0.176	0.231	0.200	0.221	0.283	**0.248***
Semeval	Proximité	0.189	0.196	**0.192***	0.036	0.038	0.037
	Degré	0.185	0.193	**0.189***	0.084	0.088	0.086
	Vecteur propre	0.185	0.193	**0.189***	0.079	0.081	0.080
	Intermédiarité	0.128	0.133	**0.131***	0.071	0.075	0.073
	Hits	0.17	0.185	**0.181***	0.079	0.081	0.080
NUS	Proximité	0.226	0.250	**0.237***	0.038	0.042	0.040
	Degré	0.233	0.257	**0.245***	0.118	0.133	0.125
	Vecteur propre	0.217	0.235	**0.226***	0.102	0.110	0.106
	Intermédiarité	0.163	0.188	**0.174***	0.106	0.122	0.114
	Hits	0.219	0.243	**0.230***	0.102	0.110	0.106
PubMed	Proximité	0.128	0.131	**0.130***	0.0167	0.020	0.018
	Degré	0.143	0.149	**0.146***	0.085	0.091	0.088
	Vecteur propre	0.140	0.145	**0.142***	0.087	0.092	0.090
	Intermédiarité	0.114	0.116	**0.115***	0.084	0.088	0.086
	Hits	0.138	0.143	**0.140***	0.087	0.092	0.090
ACM	Proximité	0.157	0.166	**0.161***	0.017	0.018	0.017
	Degré	0.164	0.175	**0.169***	0.089	0.096	0.092
	Vecteur propre	0.150	0.159	**0.155***	0.071	0.076	0.073
	Intermédiarité	0.112	0.119	**0.115***	0.074	0.078	0.076
	Hits	0.145	0.155	**0.150***	0.071	0.076	0.073

TABLEAU 2.3 – suite de la page précédente

	Méthodes	Notre			[Boudin 2013a]		
		P	R	F	P	R	F
Citeulike-180	Proximité	0.085	0.092	**0.088***	0.001	0.001	0.001
	Degré	0.127	0.135	**0.131***	0.078	0.084	0.081
	Vecteur propre	0.123	0.130	**0.127***	0.058	0.061	0.059
	Intermédiarité	0.177	0.189	**0.183***	0.119	0.121	0.120
	Hits	0.144	0.154	**0.149***	0.058	0.061	0.059

Comparaison avec les méthodes de l'état de l'art

Dans cette section, nous comparons notre approche par rapport aux autres méthodes non supervisées de l'état de l'art dont les implémentations sont disponibles. Dans cette section et dans la suite de ce manuscrit, lorsque nous parlons de "notre" méthode, il s'agit de celle avec la mesure de centralité qui fournit le meilleur résultat dans le tableau 2.3 pour chaque jeu de données. En d'autres termes, nous considérons la mesure Proximité pour Inspec et Semeval, Vecteur propre pour TALN-Archives, KDD, WWW et 500N-KPCrowd, Hits pour DUC-2001, Degré pour NUS, PubMed et ACM, et enfin Intermédiarité pour Citeulike-180.

Pour les autres méthodes de l'état de l'art, nous avons considéré Tf-Idf, KP-Miner, YAKE, TextRank, SingleRank, STPR (Single Topical PageRank), PositionRank, TopicRank et MultipartiteRank, qui sont des méthodes disponibles/implémentées dans l'outil Pke [2] [Boudin 2016]. Pour EmbedRank, nous avons pris l'implémentation disponible sur github [3]. Pour KCRank :H1 et KCRank :H1, nous les avons implémentées en suivant les instructions dans le papier de Won *et al.* [Won 2019]. Les méthodes KP-Miner et Tf-Idf utilisent tous les autres documents du jeu de données pour extraire les termes-clés d'un document alors que les autres méthodes, incluant la nôtre, n'utilisent que le document lui-même.

Pour connaître si la différence entre deux méthodes est significative, nous avons utilisé le test de Tukey avec la valeur p plus petite que 0.05. Nous avons adopté ce test parce que nous faisons face ici à une comparaison multiple, c'est-à-dire que nous devons comparer les méthodes deux par deux ; dans ce cas, utiliser des tests sans correction (comme le test de Student) amène des erreurs [Fuhr 2017].

2. `https://github.com/boudinfl/pke`, consulté le 16 Avril 2019.
3. `https://github.com/swisscom/ai-research-keyphrase-extraction`, consulté le 02 Mai 2019.

Le tableau 2.4 indique les résultats de notre méthode comparés à celles de l'état de l'art sur les cinq jeux de données contenant des documents longs ou complets. Nous constatons que la méthode KP-Miner donne les meilleurs résultats quel que soit le jeu de données en considérant les trois mesures que sont la précision (P), le rappel (R) et la f1-mesure (F). Toutefois, notre méthode obtient de bons résultats puisque nous avons le deuxième meilleur score sur trois jeux de données : Semeval, NUS et ACM. Sur le corpus PubMed, nous obtenons le quatrième meilleur score. Et enfin sur le corpus Citeulike-180, nous obtenons le troisième score en considérant la précision (P) et la f1-mesure(F), et le deuxième meilleur score en considérant le rappel (R).

En considérant les tests statistiques, il n'y a pas de différence significative entre notre méthode et la meilleure (KP-Miner) sur les corpus Semeval et Citeulike-180, alors que la différence est significative sur les corpus NUS et ACM. Sur le corpus PubMed, sur les trois méthodes de l'état de l'art qui surpassent notre méthode, seules deux sont statistiquement significativement différentes.

Le tableau 2.5 indique les résultats de notre méthode comparés à celles de l'état de l'art sur les quatre jeux de données contenant des documents courts ou résumés. En considérant la f1-mesure (F), notre méthode obtient le meilleur résultat sur le corpus Inspec, mais le sixième score sur les corpus TALN-Archives et KDD, et même le huitième score sur WWW. En considérant les tests statistiques, notre méthode ne surpasse significativement que la moitié des méthodes de l'état de l'art (six sur douze) sur le corpus Inspec. Sur les corpus TALN-Archives et KDD, seule la meilleure méthode (respectivement PositionRank et Tf-Idf) surpasse la nôtre significativement. Enfin, sur le corpus WWW, seules deux méthodes (Tf-Idf et PositionRank) offrent une amélioration statistiquement significative.

Le tableau 2.6 indique les résultats de notre méthode comparés à celles de l'état de l'art sur les deux jeux de données contenant des documents de type article de presse. En considérant les trois mesures, notre méthode obtient le deuxième meilleur score sur le corpus 500N-KPCrowd, mais le neuvième score sur le corpus DUC-2001. Sur le corpus 500N-KPCrowd, seule la différence avec la meilleure méthode (KP-Miner) est statistiquement significative alors que sur le corpus DUC-2001, seules les différences entre deux méthodes de l'état de l'art (TextRank et TopicRank) ne sont pas significatives. C'est-à-dire que sur DUC-2001, il y a sept méthodes qui surpassent la nôtre significativement.

TABLEAU 2.4 – Comparaison des résultats par rapport aux méthodes états de l'art sur les jeux de données de type papier complet. Les meilleurs résultats sont en gras. ▲ indique une amélioration significative par rapport à notre méthode avec le test de Tukey (valeur p < 0.05) alors que ▼ indique l'inverse.

Méthodes	Semeval			NUS			PubMed			ACM			Citeulike-180		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Notre méthode	.189	.196	.192	.233	.257	.245	.143	.149	.146	.164	.175	.169	.177	.189	.183
KP-Miner	**.195**	**.203**	**.199**	**.265**	**.302**	**.283**▲	**.210**▲	**.210**▲	**.210**▲	**.189**▲	**.199**▲	**.194**▲	**.203**	**.208**	**.205**
Tf-Idf	.155	.166	.160	.189▼	.218	.202▼	.178▲	.177▲	.178▲	.114▼	.119▼	.116▼	.142	.147	.144
YAKE	.151	.159	.155	.186▼	.207▼	.196▼	.136	.137	.137	.103▼	.109▼	.106▼	.180	.188	.184
MultipartiteRank	.150	.156	.153	.189▼	.201▼	.194▼	.160	.162	.161	.115▼	.118▼	.116▼	.143	.151	.147
TopicRank	.125▼	.129▼	.127▼	.161▼	.168▼	.164▼	.138	.140	.139	.098▼	.101▼	.100▼	.119▼	.124▼	.121▼
KCRank :H1	.142▼	.147▼	.144▼	.147▼	.163▼	.155▼	.089▼	.096▼	.092▼	.100▼	.103▼	.102▼	.024▼	.026▼	.025▼
KCRank :H2	.154	.159	.156	.157▼	.176▼	.166▼	.103▼	.112▼	.107▼	.109▼	.113▼	.111▼	.020▼	.020▼	.020▼
TextRank	.023▼	.024▼	.024▼	.017▼	.013▼	.015▼	.008▼	.010▼	.009▼	.008▼	.008▼	.008▼	.0▼	.0▼	.0▼
SingleRank	.031▼	.033▼	.032▼	.025▼	.023▼	.024▼	.013▼	.016▼	.014▼	.012▼	.013▼	.013▼	.002▼	.002▼	.002▼
STPR	.039▼	.040▼	.039▼	.031▼	.029▼	.030▼	.017▼	.020▼	.019▼	.015▼	.016▼	.016▼	.001▼	.001▼	.001▼
PositionRank	.075▼	.077▼	.076▼	.083▼	.090▼	.087▼	.045▼	.051▼	.048▼	.046▼	.050▼	.048▼	.011▼	.010▼	.010▼
EmbedRank	.053▼	.055▼	.054▼	.054▼	.053▼	.054▼	.029▼	.032▼	.031▼	.019▼	.021▼	.020▼	.012▼	.011▼	.012▼

TABLEAU 2.5 – Comparaison des résultats par rapport aux méthodes états de l'art sur les jeux de données de type résumé. Les meilleurs résultats sont en gras. ▲ indique une amélioration significative par rapport à notre méthode avec le test de Tukey (valeur p < 0.05) alors que ▼ indique l'inverse.

Méthodes	Inspec			TALN-Archives			KDD			WWW		
	P	R	F	P	R	F	P	R	F	P	R	F
Notre méthode	**.361**	**.421**	**.389**	.122	.164	.140	.095	.124	.108	.094	.110	.101
Tf-Idf	.118▼	.146▼	.131▼	.128	.165	.144	**.120**▲	**.152**	**.134**▲	**.133**▲	**.148**▲	**.140**▲
PositionRank	.316▼	.372▼	.342▼	**.153**▲	**.201**	**.174**▲	.112	.145	.126	.115▲	.131▲	.122▲
KP-Miner	.010▼	.012▼	.011▼	.018▼	.021▼	.019▼	.061▼	.080▼	.069▼	.057▼	.061▼	.059▼
YAKE	.127▼	.162▼	.142▼	.107	.140	.121	.054▼	.069▼	.061▼	.072▼	.079▼	.075▼
TextRank	.340	.398	.367	.074▼	.099▼	.085▼	.041▼	.051▼	.046▼	.048▼	.057▼	.052▼
SingleRank	.337	.397	.364	.116	.156	.133	.074	.097	.084	.078	.092	.084
STPR	.338	.398	.365	.120	.160	.137	.082	.104	.092	.085	.099	.091
TopicRank	.280▼	.320▼	.299▼	.124	.160	.140	.088	.110	.097	.103	.112	.108
MultipartiteRank	.288▼	.337▼	.310▼	.135	.174	.152	.098	.123	.109	.113	.123	.117
KCRank :H1	.351	.406	.376	.133	.177	.152	.098	.127	.111	.097	.114	.105
KCRank :H2	.357	.414	.383	.137	.182	.156	.100	.129	.113	.097	.113	.105
EmbedRank	.335	.389	.360	.119	.157	.135	.088	.113	.099	.099	.113	.105

TABLEAU 2.6 – Comparaison des résultats par rapport aux méthodes états de l'art sur les jeux de données de type article de presse. Les meilleurs résultats sont en gras. ▲ indique une amélioration significative par rapport à notre méthode avec le test de Tukey (valeur p < 0.05) alors que ▼ indique l'inverse.

Méthodes	500N-KPCrowd			DUC-2001		
	P	R	F	P	R	F
Notre méthode	0.248	0.266	0.257	0.176	0.231	0.200
MultipartiteRank	**0.263**	**0.283**	**0.273**	0.222▲	0.288▲	0.250▲
KCRank :H1	0.214	0.245	0.228	**0.282**▲	**0.360**▲	**0.316**▲
KCRank :H2	0.215	0.245	0.229	0.280▲	0.357▲	0.313▲
Tf-Idf	0.213	0.221	0.217	0.090▼	0.121▼	0.103▼
KP-Miner	0.057▼	0.051▼	0.054▼	0.065▼	0.084▼	0.074▼
YAKE	0.238	0.264	0.250	0.104▼	0.134▼	0.117▼
TextRank	0.181▼	0.213	0.196	0.166	0.212	0.186
SingleRank	0.218	0.246	0.231	0.224▲	0.286▲	0.251▲
STPR	0.216	0.243	0.228	0.234▲	0.299▲	0.262▲
PositionRank	0.222	0.247	0.234	0.256▲	0.327▲	0.287▲
TopicRank	0.236	0.254	0.245	0.205	0.265	0.231
EmbedRank	0.226	0.248	0.237	0.274▲	0.352▲	0.308▲

2.5 Conclusion

Dans ce chapitre, nous proposons une méthode non supervisée d'extraction de termes-clés à base de graphes. Beaucoup de méthodes non supervisées sont à base de graphes dans la littérature du fait de leur bonne performance, leur simplicité et la facilité de leur implantation et elles sont aussi les plus courantes et suffisamment diversifiées. Dans nos travaux, nous proposons d'améliorer la méthode proposée par Boudin *et al.* [Boudin 2013a] parce qu'elle utilise des algorithmes simples pour l'ordonnancement des nœuds qui sont les mesures de centralité et offre des performances équivalentes aux autres méthodes à base de graphes.

Les méthodes à base de graphes de l'état de l'art comportent des faiblesses :

(1) la construction du graphe de mots est basée sur la co-occurrence qui ne capture pas très bien la relation sémantique entre deux mots.

(2) l'attribution de scores aux termes-clés candidats est en fonction de leur longueur ce qui favorise souvent les candidats les plus longs ou les plus courts.
(3) le calcul des scores des termes-clés candidats est en fonction des scores des mots qui les composent ce qui favorise la présence de chevauchements des termes-clés, c'est-à-dire que certains candidats sont des sous-chaînes d'autres candidats.
(4) la construction du graphe de mots basée sur la co-occurrence entraîne aussi la perte d'informations qui peuvent être très importantes dans le choix des termes-clés telles que la fréquence ou la position des termes-clés candidats dans le document.

Pour résoudre ces problèmes, nous avons proposé d'utiliser le plongement de mots lors de la construction du graphe de mots ; celui-ci capture en effet les relations entre les mots. Puis nous avons adopté une modification de la moyenne harmonique [Yeom 2019] pour calculer les scores des termes-clés candidats pour résoudre le problème (2). Enfin pour résoudre les problèmes (3) et (4), nous avons recalculé les scores des termes-clés candidats en utilisant la méthode modifiée C-value proposée par Yeom *et al.* [Yeom 2019].

Afin d'évaluer notre approche, nous avons utilisé onze collections de données dont cinq contenant des documents longs ou complets, quatre contenant des documents courts ou résumés et enfin deux contenant des documents de type article de presse.

Comme résultats, nous pouvons dire que les modifications que nous avons apportées à l'approche de Boudin *et al.* [Boudin 2013a] fonctionnent mieux sur les documents longs que sur les documents courts et les documents de type article de presse. Cependant, sur un des corpus qui contient des documents courts, notre approche marche aussi très bien. En effet, sur les cinq corpus de type long, la différence sur la f1-mesure (F) entre notre approche et celle de Boudin *et al.* est statistiquement significative avec un test de Student (valeur $p < 0.05$). Sur les quatre corpus de type court, notre méthode fournit des performances statistiquement supérieures sur un jeu de données, mais des performances statistiquement inférieures sur les trois autres. Enfin, sur les corpus de type article de presse, sur le premier corpus, la différence n'est pas statistiquement significative alors que sur le second, la performance de notre méthode est statistiquement inférieure.

Comparée à douze autres méthodes non supervisées de l'état de l'art, notre méthode obtient les mêmes résultats que les meilleures méthodes sur deux des cinq corpus de type long, le deuxième résultat sur deux autres corpus et le troisième résultat sur le dernier corpus. En considérant les corpus de type court, notre méthode fait partie des meilleures méthodes sur un des corpus, obtient le deuxième résultat sur deux corpus et enfin le troisième résultat sur le dernier corpus. Enfin, sur les deux corpus de

type article de presse, notre approche obtient le deuxième meilleur résultat sur l'un et seulement le huitième sur l'autre.

Malgré ces performances, notre méthode présente aussi quelques faiblesses. Comme nous avons utilisé un modèle pré-entraîné de plongement de mots pour la langue anglaise, notre méthode est donc difficilement généralisable à d'autres langues à moins de trouver ou d'entraîner un modèle sur la langue à traiter. Aussi, le temps de traitement des longs documents est un peu long comparé à d'autres méthodes comme KCRank [Won 2019] mais ce problème est général à toutes les méthodes à base de graphes.

Comme travaux futurs, nous prévoyons d'explorer d'autres méthodes de plongement de mots/termes/phrases ou de représentation de texte telles que BERT [Devlin 2019]. Nous aimerions aussi explorer d'autres méthodes d'extraction de termes-clés candidats qui garantissent l'extraction de bons candidats. Sur l'ensemble de données Semeval par exemple, nous avons extrait beaucoup de candidats qui contiennent des symboles et ne sont à priori pas de bons candidats.

Le travail et les résultats présentés dans ce chapitre ont donné lieu aux publications suivantes :

1. Josiane Mothe, Faneva Ramiandrisoa et Michael Rasolomanana. Automatic keyphrase extraction using graph-based methods. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018, pages 728–730, 2018.
2. Josiane Mothe, Michel Rajoelina, Faneva Ramiandrisoa et Hary Razakasoa. Intégration des plongements de mots dans les méthodes, supervisées et non supervisées, d'extraction automatique de mots clés. VSST 2018.
3. Josiane Mothe et Faneva Ramiandrisoa. Extraction automatique de termes-clés : Comparaison des méthodes non supervisées de la littérature. In CORIA 2016 - Conférence en Recherche d'Informations et Applications - 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, pages 315–324, 2016.

Chapitre 3

Détection de la dépression

Sommaire

Résumé.

Selon l'Organisation Mondiale de la Santé (OMS), le nombre de personnes atteintes de troubles mentaux dans le monde est en augmentation [Organization 2017] et la dépression est l'une des formes les plus courantes de ces troubles, avec l'anxiété. Selon cette même organisation, le nombre de personnes dans le monde souffrant de dépression est d'environ 300 millions. La détection de ce trouble est donc devenue cruciale et constitue un défi pour la santé individuelle et publique. De nombreuses études et recherches ont été consacrées à ce défi et il existe aussi des compétitions dont le but est de créer des systèmes permettant de résoudre ces défis. L'une de ces compétitions est la tâche eRisk dont l'objectif est de détecter le plus tôt possible les signes de la dépression dans les textes des utilisateurs de réseaux sociaux. Notre travail propose une solution pour résoudre le problème abordé dans la tâche eRisk. Pour cela, nous proposons d'utiliser des classifieurs classiques, tel que la régression logistique, utilisant en entrée : (a) des vecteurs de caractéristiques et (b) des vecteurs basés sur le plongement de phrases ; ces vecteurs sont construits à partir des publications des utilisateurs. Pour l'évaluation de notre approche, nous avons utilisé deux collections de données sur la dépression qui sont les jeux de données des deux éditions de eRisk (2017 et 2018). L'édition 2018 de la tâche eRisk propose aussi de résoudre le problème de la détection au plus tôt de l'anorexie et nous avons décidé d'utiliser notre méthode pour résoudre ce problème afin de voir sa portabilité sur des problèmes autres que la dépression. Nous avons observé que les modèles basés sur les caractéristiques sont très performants lorsque la mesure de précision est considérée, que cela soit pour la détection de la dépression ou pour la détection de l'anorexie. Le modèle utilisant le plongement de phrases, quant à lui, est plus performant lorsque l'on mesure la détection au plus tôt ($ERDE_{50}$) et le rappel. Nous avons aussi obtenu de bons résultats par rapport à l'état de l'art : meilleurs résultats sur la précision et $ERDE_{50}$ pour la détection de la dépression, et sur la précision et le rappel pour la détection de l'anorexie.

3.1 Introduction

Selon l'Organisation Mondiale de la Santé (OMS), le nombre de personnes atteintes de troubles mentaux dans le monde est en augmentation [Organization 2017]. Selon cette organisation, les troubles mentaux les plus courants sont la dépression et l'anxiété.

Globalement, le nombre de personnes dans le monde souffrant de dépression est d'environ 300 millions ; cela correspond à une augmentation de plus de 18 % entre 2005 à 2015 [1]. L'OMS a aussi constaté que la dépression touche plus les femmes que les hommes. La figure 3.1 montre le fardeau mondial de la dépression dont près de la moitié de ces personnes vivent dans les régions de l'Asie du Sud-Est et du Pacifique occidental.

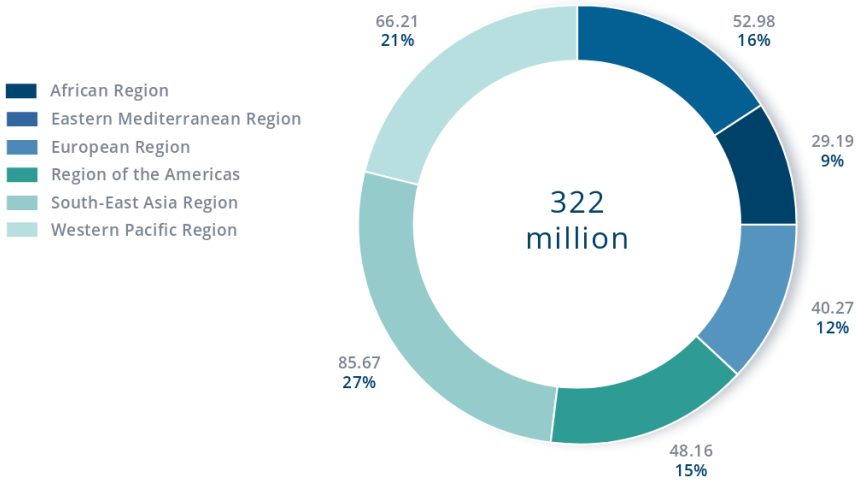


FIGURE 3.1 – Cas de trouble dépressif par région en 2017 selon l'OMS. [Organization 2017]

D'après une autre étude publiée dans la revue BMC Medicine, qui a analysé des données fournies par l'OMS sur les taux de dépression de 18 pays [2], la France est le

1. https://www.la-depression.org/comprendre-la-depression/la-depression-en-chiffre/, consulté le 13 Novembre 2018

2. Les 18 pays sont : France, Japon, Allemagne, Italie, Israël, Espagne, Belgique, Nouvelle Zélande, Pays Bas, Etats-Unis, Chine, Mexique, Inde, Afrique du Sud, Liban, Colombie, Ukraine, Brésil.

pays le plus touché avec un taux de 21 % suivie des Etats-Unis (19,2 %)[3]. En effet, en France, on estime que près d'une personne sur cinq a souffert ou souffrira d'une dépression au cours de sa vie. Ainsi, la détection de ce trouble est cruciale et constitue un défi pour la santé individuelle et publique.

De nombreuses études et recherches ont été consacrées à ce défi [France 2000, Low 2011, Ozdas 2004]. Bien qu'il existe des facteurs cliniques qui peuvent aider à la détection précoce des patients à risque [Sagen 2010], il existe aussi des usages linguistiques spécifiques aux états dépressifs [Pennebaker 2003, Rude 2004]. En effet, des chercheurs ont constaté que la dépression était associée à des schémas linguistiques spécifiques tels que l'usage excessif de pronoms personnels, du passé ou des émotions négatives. Les écrits d'une personne peuvent donc être utilisés pour capturer son état psychologique.

Au cours de ces dernières années, l'émergence des réseaux sociaux tels que Facebook, Twitter ou Reddit a permis aux personnes de partager leurs expériences personnelles, leurs idées ou leurs pensées de manière plus simple. Selon l'auteur dans [Kumar 2018a], les gens préfèrent en fait s'exprimer en ligne plutôt que hors ligne. Ce phénomène a généré beaucoup de données qui sont l'opportunité de nombreux thèmes de recherche, par exemple dans le domaine médical [Dalloux 2020].

De plus, une étude faite par Marriott et Buchanan [Marriott 2014] montre qu'il n'y a pas de différence significative entre la personnalité en ligne d'un individu et sa personnalité hors ligne en termes d'authenticité. Il devient alors possible d'étudier les écrits de ces utilisateurs afin d'essayer de détecter les utilisateurs dépressifs en s'appuyant sur des indicateurs linguistiques. La plupart des approches de la littérature sur la détection de la dépression dans les réseaux sociaux utilisent des méthodes d'apprentissage supervisé entraînées sur des jeux de données annotés manuellement. Plusieurs groupes de caractéristiques ont été utilisés comme l'utilisation des émoticônes [Wang 2013] ou de jurons [Schwartz 2014], l'heure de publication sur le réseau social [Choudhury 2013] et les thèmes mentionnés [Resnik 2015].

Dans le travail de recherche que nous rapportons dans ce chapitre, nous avons utilisé une méthode d'apprentissage supervisée pour la détection de la dépression, plus précisément la détection au plus tôt de la dépression. Nous avons plusieurs textes de l'utilisateur à disposition, ordonnés par ordre chronologique, et l'objectif est de détecter si cet utilisateur est dépressif, en utilisant le moins de textes possible (du plus ancien au plus récent). En d'autres termes, nous obtenons les textes en flux et le flux s'arrête

3. `http://www.doctissimo.fr/psychologie/news/la-france-pays-le-plus-touche-par-la-depression`, consulté le 13 Novembre 2018

lorsque l'utilisateur est classé dépressif (ou lorsqu'il n'y a plus de nouvelle donnée). Ce travail a été validé sur les jeux de données de la tâche eRisk, des éditions 2017 et 2018. eRisk est une tâche qui se focalise sur la prédiction précoce des risques sur internet dont la dépression (voir Section 3.4 pour plus de détails sur la tâche). Ce travail propose donc une solution pour résoudre le problème abordé dans la tâche eRisk. L'édition 2018 de la tâche eRisk propose aussi de résoudre le problème de la détection au plus tôt de l'anorexie. Nous avons aussi utilisé notre méthode pour résoudre ce problème afin de voir sa portabilité sur des problèmes autres que la dépression.

Le chapitre est organisé comme suit : la Section 3.2 présente l'état de l'art sur les méthodes d'identification des signes de dépression provenant des médias sociaux. La Section 3.3 détaille notre méthode pour résoudre les problèmes abordés durant la tâche eRisk des éditions 2017 et 2018, c'est-à-dire la détection au plus tôt de la dépression. La Section 3.4 présente les jeux de données des deux éditions de la tâche eRisk (2017 et 2018), que nous avons utilisés pour évaluer notre approche et décrit aussi la tâche eRisk. La Section 3.5 présente les résultats de notre méthode. La Section 3.6 présente les résultats de notre méthode pour la détection de l'anorexie. Enfin la Section 3.7 conclut ce chapitre.

3.2 État de l'art

Dans cette section, nous discutons des travaux sur l'identification des signes de dépression provenant des médias sociaux.

La majorité des travaux sur l'identification de la dépression dans les médias sociaux tentent de rendre le processus de diagnostic automatique. Plusieurs de ces travaux sont majoritairement basés sur des méthodes d'apprentissage supervisé.

Certaines méthodes collectent des éléments statistiques en quantifiant l'activité sur les plates-formes de réseaux sociaux, l'isolement social, le nombre d'amis, le réseau d'utilisateurs basé sur les intérêts des utilisateurs ou des connexions mutuelles, etc. D'autres méthodes utilisent les techniques de traitement du langage naturel, elles sont essentiellement basées sur l'analyse linguistique et sémantique des textes comme l'analyse des émotions exprimées, la structure des phrases, etc.

De Choudhury et al. [Choudhury 2014] définissent plusieurs mesures pour caractériser les différences entre les mères atteintes de dépression post-partum (DPP : trouble de l'humeur pouvant affecter les mères après l'accouchement) et celles qui ne le sont pas. Pour constituer leurs données, les auteurs ont fait appel à des mères qui ont donné naissance dans les 9 derniers mois et possédant un compte Facebook. Pour identifier les participantes qui sont réellement atteintes de DPP, ils ont considéré l'auto-déclaration

et les scores des participantes au questionnaire PHQ-9 (Patient Health Questionnaire-9), un instrument psychométrique très utilisé pour détecter la dépression. À la fin, ils ont gardé 165 participantes dont 28 sont atteintes de DPP et 137 non atteintes. Pour construire leur jeu de données, ils ont collecté les données produites par ces participantes sur Facebook, telles que les photos, publications, commentaires, mention j'aime, etc., pendant les périodes prénatales (50 semaines) et postnatales (10 semaines). À partir du jeu de données, ils ont pu définir 49 caractéristiques (mesures) qui ont été regroupées en 4 catégories : *caractéristiques de l'utilisateur* (regroupe les caractéristiques qui mesurent l'activité sur le réseau social comme le nombre de publications par jour), *capital social* (regroupe les caractéristiques qui mesurent l'interaction avec les autres comme le nombre de mentions *j'aime* des publications des amis), *caractéristiques du contenu* (regroupe les caractéristiques qui sont calculées à partir du contenu des publications comme l'analyse des émotions) et *style linguistique* (regroupe les caractéristiques qui mesurent les changements de comportement basés sur l'utilisation de styles linguistiques dans les publications). Les auteurs ont ensuite fait des analyses statistiques pour savoir quelles sont les caractéristiques qui permettent de mieux différencier les deux groupes. Ils ont constaté que 35 des 49 caractéristiques permettent de distinguer significativement (t-test), les deux groupes. Après ces analyses, ils ont développé plusieurs modèles de régression, avec différentes combinaisons de caractéristiques et des données démographiques, pour prédire si une mère risque d'avoir une DPP. Ils ont d'abord évalué leurs modèles seulement sur les données collectées pendant la période prénatale puis ils y ont ajouté les données collectées pendant 1 mois durant la période postnatale. Sur les données prénatales, le modèle utilisant toutes les caractéristiques donne le meilleur résultat, mais leur meilleur modèle est celui basé sur les données prénatales combinées avec 1 mois de données postnatales qui n'est malheureusement pas détaillé dans leurs travaux. Dans nos travaux, nous nous sommes inspirés ou avons adapté certaines caractéristiques dans les groupes *caractéristiques du contenu* et *style linguistique*.

Dans un autre de leur travail, De Choudhury *et al.* [Choudhury 2013] définissent plusieurs caractéristiques pour caractériser les comportements des personnes atteintes de la dépression. Les auteurs ont utilisé le crowdsourcing pour obtenir des utilisateurs et pour mesurer leur niveau de dépression, le questionnaire CES-D (Center for Epidemiologic Studies Depression Scale) et le BDI (Beck Depression Inventory) ont été utilisés. Au total, 1 583 personnes ont répondu à leur appel dont 637 ont accepté de donner l'accès à leurs comptes Twitter. Après avoir filtré les participants, 476 utilisateurs ont été retenus, dont 171 ont eu un score CES-D élevé et 305 ont eu un score faible ou aucun signe de dépression. Dans ce travail, les données ont été collectées sur

Twitter concernant les données des participants et 43 caractéristiques ont été définies et groupées dans les catégories suivantes : *engagement*, *graphe social égocentrique (réseau égocentrique)*, *émotion*, *style linguistique* et *langage de la dépression*. En plus de ces caractéristiques calculées à partir des données collectées sur Twitter, 4 caractéristiques démographiques ont été considérées : *âge*, *genre*, *niveau d'éducation* et *revenu*. Avec ces caractéristiques, plusieurs analyses ont permis de distinguer les comportements entre personnes dépressives et non dépressives. Les auteurs ont constaté que celles atteintes de la dépression font moins d'activité sociale, manifestent plus d'émotions négatives, portent une très grande attention à elles-mêmes, montrent une augmentation des préoccupations relationnelles et médicales, et expriment plus de pensées religieuses. Ils ont aussi observé que même si leurs réseaux égocentriques sont petits, les utilisateurs dépressifs semblent appartenir à des réseaux étroitement (ou très) groupés et sont généralement très liés aux contacts dans leur réseau. Une fois ces analyses faites, les auteurs ont créé plusieurs modèles, en utilisant différentes combinaisons de caractéristiques et différentes méthodes de classification pour prédire si un utilisateur est dépressif ou non. Ils ont utilisé la validation croisée (k=10) pour évaluer leurs modèles et ont constaté que le classifieur SVM entraîné avec des caractéristiques, de dimensions réduites avec la méthode d'analyse en composantes principales (ACP), fournit le meilleur résultat avec une exactitude de 70 % et une précision de 0.74. Dans nos travaux, nous nous sommes inspirés ou avons adapté certaines caractéristiques dans les groupes *émotion*, *engagement* et *langage de dépression*.

Trotzek *et al.* [Trotzek 2018, Trotzek 2017] ont participé deux fois à la tâche eRisk (l'année 2017 et 2018). L'objectif de la tâche est de détecter le plus tôt possible les risques reliés à la santé et à la sécurité sur internet (plus de détails sont donnés dans la section 3.4.2). Lors de la première édition, la tâche consistait à détecter le plus tôt possible la trace d'une dépression à partir des textes des utilisateurs. Le jeu de données utilisé durant la tâche est composé des textes des utilisateurs, ordonnés dans l'ordre chronologique, postés sur le forum reddit. Pour prendre en compte la dimension de temporalité, le jeu de données a été divisé en 10 partitions (partitions) ; chaque partition contient 10 % des textes des utilisateurs (détail dans la section 3.4.1). L'objectif de la tâche est de prédire si les utilisateurs sont dépressifs en utilisant le moins de partitions possible. Trotzek *et al.* ont développé 5 modèles pour la détection de dépression. Tous les modèles utilisent les méta-informations linguistiques extraites des textes de chaque utilisateur et les combinent avec des classifieurs basés sur un sac de mots, un vecteur de paragraphe, une analyse sémantique latente (LSA) et des réseaux de neurones récurrents (RNN). Leur modèle, basé sur le sac de mots, a donné les meilleurs résultats en considérant la f1-mesure et le modèle basé sur le vecteur de paragraphe a donné les

meilleurs résultats en considérant la précision et la mesure $ERDE_5$ (mesure définie pour la tâche eRisk, voir section 3.4 pour plus de détails). Lors de la seconde édition, qui est composée de deux sous-tâches : la détection de la dépression et la détection de l'anorexie, les auteurs ont utilisé quatre modèles d'apprentissage automatique et un modèle d'ensemble combinant les résultats obtenus à partir des quatre modèles [Trotzek 2018]. Alors que deux de leurs modèles d'apprentissage automatique sont basés sur CNN (réseau de neurones convolutionnels), les deux autres sont basés sur des caractéristiques calculées à partir du texte de l'utilisateur : un modèle basé sur des méta-données linguistiques et des sacs de mots, et un modèle basé seulement sur les sacs de mots. Lors du défi eRisk 2018, ils ont obtenu les meilleures performances selon les mesures $ERDE_{50}$ et f1-mesure dans les deux tâches (dépression et anorexie). Pour la détection de la dépression, leur modèle basé, uniquement sur les sacs de mots, a donné les meilleurs résultats et pour la détection de l'anorexie, il s'agit de la combinaison du modèle basé sur les sacs de mots et les méta-données et des deux modèles basés sur CNN. Dans nos travaux, nous allons utiliser leur modèle utilisant le vecteur de paragraphe et aussi 4 caractéristiques qu'ils ont utilisées.

Après la participation à l'édition 2017 de la tâche eRisk, Trotrzek *et al.* [Trotzek 2020] ont étendu leur recherche et ont amélioré leurs résultats (sur le jeu de données de eRiks 2017). Dans ce nouveau travail, ils ont repris les caractéristiques qu'ils ont présentées et décrites précédemment dans [Trotzek 2017] et ont développé un modèle utilisant un classifieur de type régression logistique. Ils ont aussi développé plusieurs modèles basés sur des CNN. La différence entre les différents modèles basés sur les CNN est le modèle de plongement de mots qu'ils ont utilisé. Enfin, ils ont proposé plusieurs modèles d'ensemble qui sont des combinaisons de deux modèles : l'un utilisant les caractéristiques et l'autre basé sur les CNN (différent selon le plongement de mots utilisé). Sur le jeu de données de la tâche eRisk 2017, ils ont obtenu le meilleur résultat de l'état de l'art sur les mesures $ERDE_{5}$ et $ERDE_{50}$ avec un de leurs modèles d'ensemble où le modèle basé sur CNN utilise fastText [Joulin 2017], entraîné sur des articles Wikipédia, comme plongement de mots.

Funez *et al.* [Funez 2017b, Funez 2018] ont aussi participé aux deux éditions de eRisk (2017 et 2018). Pour l'édition 2017, ils ont proposé un modèle combinant trois modèles : un modèle basé sur la représentation de documents en sac de mots, un modèle qui utilise Variation Temporelle des Termes ou Temporal Variation of Terms (TVT) pour la représentation sémantique de documents et enfin un modèle d'arbre de décision obtenu en sélectionnant les mots ayant les gains d'information les plus élevés. TVT est une méthode de représentation sémantique de documents qui considère la variation du vocabulaire à travers le temps comme un espace conceptuel. La méthode

s'appuie sur des techniques d'analyse sémantique concise (CSA : Concise Semantic Analysis). La CSA représente les termes en vecteur dans un espace de concepts qui est égal ou proche des labels (classes) ; et un document est le centroïde des vecteurs de termes qui le composent. L'idée principale de la méthode de représentation TVT est d'utiliser les informations temporelles pour obtenir un espace conceptuel étendu (plus de détails sur TVT peut être trouvé dans le papier [Funez 2017a]). Leur modèle a obtenu le meilleur résultat durant la tâche en considérant la mesure ERDE_{50}. Juste après leur participation à la tâche eRisk 2017, Funez *et al.* [Funez 2017a] ont refait des expérimentations en utilisant uniquement leur méthode de représentation sémantique de documents TVT. Ils ont obtenu de meilleurs résultats, par rapport à leur participation à eRisk 2017, en considérant les mesures ERDE_5 et ERDE_{50} en utilisant, respectivement, le classifieur forêts d'arbres de décision et le classifieur Naïf de Bayes. Pour l'édition 2018 de la tâche eRisk, ils ont proposé deux approches différentes : Variation Temporelle Flexible des Termes ou Flexible Temporal Variation of Terms (FTVT) qui est une amélioration de TVT, et Classification Séquentielle Incrémentale ou Sequential Incremental Classification (SIC). Pour leur participation, ils ont soumis cinq modèles dont trois sont des variantes [4] de l'approche FTVT et deux sont des variantes[4] de l'approche SIC. Ils ont obtenu les meilleurs résultats en considérant la mesure ERDE_5 pour la détection de la dépression et de l'anorexie avec l'approche FTVT. Avec l'approche SIC, ils ont obtenu le meilleur résultat en considérant la mesure Rappel pour la détection de l'anorexie.

Burdisso *et al.* [Burdisso 2019b] ont proposé une nouvelle méthode d'apprentissage supervisée, appelée SS3, pour la classification de texte qui permet de traiter les problèmes de détection au plus tôt ou précoce. La conception de SS3 vise à traiter trois principaux aspects des problèmes de détection au plus tôt : la classification incrémentale des données séquentielles, la prise en charge de la classification précoce et l'explicabilité. Durant la phase d'entraînement et pour chaque catégorie donnée, SS3 construit un dictionnaire de mots, avec leur fréquence, pour chaque catégorie. Ensuite, à partir de ces fréquences de mots, pendant la phase de classification, SS3 calcule une valeur pour chaque mot à l'aide d'une fonction $gv(w, c)$ pour évaluer la relation entre un mot w et une catégorie c. Ensuite une version vectorielle de gv, appelée vecteur de confiance, est définie : $\overrightarrow{gv(w)} = (gv(w, c_0), gv(w, c_1)$ où $c_i \in C$ et et C désigne l'ensemble de toutes les catégories. Pour une classification, SS3 divise d'abord l'entrée (par exemples des documents ou des textes) en blocs (par exemple des paragraphes), puis chaque bloc est à nouveau divisé en blocs plus petits jusqu'à ce que les mots soient atteints. Ensuite, SS3 calcule le vecteur de confiance $\overrightarrow{gv}$ pour chaque mot, puis ces vec-

4. variante ici signifie que seul les paramètres de l'approche sont différents

teurs de confiance sont réduits au moyen d'un opérateur (par exemples la somme, le maximum, etc.) pour générer les vecteurs de confiance du bloc supérieur. Ce processus de réduction se propage récursivement jusqu'aux blocs de niveau supérieur et jusqu'à ce qu'un seul vecteur de confiance soit généré pour l'entrée. Enfin, la classification est effectuée sur la base des valeurs de ce vecteur de confiance unique (par exemple en sélectionnant la catégorie ayant la valeur de confiance la plus élevée). Ils ont testé SS3 pour résoudre le problème à traiter de la tâche eRisk 2017. Ils ont obtenu le meilleur résultat en considérant la mesure ERDE_5 par rapport aux participants de la tâche. Dans le même travail, ils ont proposé $\text{SS3}^{\triangle}$, qui est différent de SS3 sur la politique de classification. Avec $\text{SS3}^{\triangle}$, ils ont obtenu le meilleur résultat en considérant la mesure ERDE_{50} par rapport aux participants de la tâche eRisk 2017.

Dans un autre de leur travail, Burdisso *et al.* étend l'approche SS3 [Burdisso 2019b] et propose $\tau-$SS3 [Burdisso 2019a]. $\tau-$SS3 reconnaît dynamiquement des motifs/patrons utiles sur les flux de texte, c'est-à-dire qu'il peut apprendre et reconnaître des N-grammes de longueur variable. Cette nouvelle méthode a été testée sur les deux éditions de la tâche eRisk (2017 et 2018) sur la détection de la dépression et a fourni de meilleurs résultats que SS3. Par rapport aux résultats des autres participants, $\tau-$SS3 a obtenu les meilleurs résultats en considérant les mesures ERDE_5 et ERDE_{50} sur l'édition 2017 et le meilleur en considérant la mesure ERDE_{50} sur l'édition 2018.

Dans la section suivante, nous allons présenter notre approche pour résoudre le problème la détection au plus tôt de la dépression.

3.3 Détection au plus tôt de la dépression

Nous avons utilisé une méthode d'apprentissage supervisée pour la détection de la dépression, plus précisément, nous avons utilisé des classifieurs classiques comme la régression logistique. Avec les classifieurs classiques, les données en entrée doivent être des réels ou des vecteurs contenant des nombres réels.

Cependant, ces classifieurs classiques ne sont pas conçus pour résoudre les problèmes de détection au plus tôt alors que nous traitons ce genre de problèmes dans ce travail. Les problèmes de détection au plus tôt intègrent une dimension temps lors de la classification, en d'autres termes, ils utilisent des flux de données comme entrée. C'est-à-dire qu'à un instant t donné, seule une partition des données est disponible pour la classification et plus on avance dans le temps, plus il y a de données à disposition. L'objectif est d'avoir les meilleurs résultats en utilisant un minimum de données, c'est-à-dire le plus rapidement possible dans le temps.

Pour permettre aux classifieurs classiques de traiter le problème auquel nous faisons face dans ce travail, nous adoptons un processus de classification où nous intégrons la dimension temps en dehors des classifieurs. Ce processus consiste premièrement à entraîner des classifieurs sans tenir compte de la dimension temps, en d'autres termes, utiliser toutes les données disponibles dans le corpus d'entraînement. Ensuite, on utilise ces classifieurs entraînés pour la détection au plus tôt de la dépression sur les données de test. Plus précisément, à un instant t et pour un utilisateur donné, un classifieur classe l'utilisateur en utilisant seulement les données disponibles à cet instant. Si l'utilisateur est classé comme dépressif alors le processus de classification s'arrête, sinon le classifieur classe à nouveau l'utilisateur en utilisant les données mises à disposition au temps t+1. Ce processus est répété et à la fin, si l'utilisateur n'est toujours pas classé par le classifieur comme dépressif, alors il est considéré comme non dépressif.

Dans ce travail, nous avons comme données des flux de textes qui sont des publications des utilisateurs. Pour savoir si un utilisateur est dépressif ou non à un instant t, nous avons un nombre N ($\geqslant 1$) de ses textes à disposition. Comme entrée des classifieurs, nous construisons des vecteurs contenant des nombres réels où chaque vecteur représente un utilisateur. Un vecteur utilisateur est construit à partir de toutes les publications disponibles dudit utilisateur à l'instant t dans le processus de la classification. Ainsi, le vecteur de l'utilisateur est différent à chaque instant t et c'est ce vecteur utilisateur qui est donné en entrée aux classifieurs pour la classification.

Pour construire le vecteur utilisateur à partir de N textes, nous avons utilisé deux techniques : la représentation de l'utilisateur par des caractéristiques et la représentation de l'utilisateur par le plongement de mots. Nous tenons à préciser que nous travaillons sur des textes écrits en anglais.

3.3.1 Représentation de l'utilisateur par des caractéristiques

Nous représentons les utilisateurs par des vecteurs de traits/caractéristiques, où chaque valeur dans le vecteur correspond à une caractéristique. Ces caractéristiques sont calculées à partir des publications des utilisateurs. Une caractéristique peut être calculée de deux façons : soit il s'agit de la moyenne des valeurs de la caractéristique calculées pour chaque publication de l'utilisateur concerné, soit il s'agit de la valeur calculée à partir d'un seul texte qui est la concaténation de ses publications.

Dans la suite de cette section, nous présentons les caractéristiques que nous avons extraites, ainsi que leurs détails de calcul. Extraites et/ou inspirées par les travaux dans la littérature, nous explorons d'abord le sac de mots. Ensuite, nous divisons les caractéristiques en sept groupes en fonction de leur nature. Pour le premier groupe de caracté-

ristiques, il s'agit du style de langue adopté par un utilisateur, par exemple l'utilisation de ponctuation spéciale ou de mots en majuscules ; pour le second groupe, il s'agit du comportement de l'utilisateur : la longueur des commentaires, l'habitude de poster en pleine nuit, etc. ; le troisième groupe traite de l'aspect psychologique : d'après la littérature, les utilisateurs dépressifs ont tendance à être plus préoccupés par eux-mêmes (utilisation plus fréquente de pronoms personnels par exemple) ; pour le quatrième groupe, il s'agit de la réminiscence : les personnes dépressives parlent davantage du passé, car elles sont davantage projetées sur ce qu'elles ont fait que sur l'avenir et ce qu'il faut faire ; pour le cinquième groupe, il s'agit de la mention des antidépresseurs et des symptômes associés à la dépression : le dépressif parlerait de sa thérapie ; pour le sixième groupe, les utilisateurs dépressifs ont tendance à exprimer un sentiment plus négatif et une émotion dépressive vis-à-vis d'une cible qu'ils/elles décrivent ; enfin, le dernier groupe concerne l'analyse textuelle des publications de l'utilisateur comme la mesure de la lisibilité d'un texte. Dans ce qui suit * indique les caractéristiques que nous avons proposées par rapport à celles qui existent dans la littérature.

Dans ce qui suit, la fréquence normalisée correspond à la division de la fréquence par le nombre total de mots dans tous les textes de l'utilisateur.

3.3.1.1 Sac de mots

Nous allons commencer avec le sac de mots qui présente un avantage évident : facile à mettre en œuvre, à comprendre et fonctionne bien dans de nombreux cas.

Le modèle par sac de mots est une représentation simple utilisée en TAL et en RI. Dans ce modèle, un texte, une phrase ou un document, est représenté comme le sac de ses mots, sans tenir compte de la syntaxe. Le modèle de sac de mots est couramment utilisé dans les méthodes de classification de documents où l'occurrence de chaque mot est utilisée comme caractéristique.

Avant d'extraire tous les mots fréquents et importants, nous avons d'abord supprimé tous les mots vides du texte. Ensuite, nous avons extrait les 50 uni-grammes les plus fréquents, en fonction de leur fréquence, à partir des écrits/textes des utilisateurs dépressifs. Nous avons ensuite répété ce processus pour obtenir les 50 bi-grammes et des 50 tri-grammes les plus fréquents. Nous avons ensuite suivi le même processus pour obtenir les 50 uni-grammes, bi-grammes et tri-grammes qui ont la plus grande valeur de Tf-Idf.

Nous avons constaté qu'avec les 50 uni-grammes extraits en fonction de leur fréquence, nous obtenions le meilleur résultat, en utilisant l'algorithme des forêts d'arbres de décision, pour détecter les personnes dépressives par rapport aux autres N-

grammes. Nous avons utilisé la mesure précision pour mesurer la performance parce que nous avons voulu retenir uniquement les N-grammes qui permettent de bien détecter la dépression. Ensuite, parmi ces 50 uni-grammes, nous n'en avons retenu que 18 dans la mesure où en n'utilisant que ces 18 uni-grammes, nous avons obtenu un résultat comparable à celui obtenu avec 50 uni-grammes. Pour la sélection des uni-grammes les plus pertinents, nous avons utilisé la méthode "Chi-squared Ranking Filter".

Ces 18 uni-grammes ont été sélectionnés sur un corpus de textes concernant la dépression (jeu d'entraînement de eRisk 2017 sur la dépression, voir section 3.4), ils ne sont donc ni généralisables ni extensibles à une tâche de détection automatique sauf si cette dernière est très liée à la dépression. Ils sont peut-être aussi dépendants du corpus dont on les a extraits (une hypothèse qu'il reste à vérifier).

Chaque publication de l'utilisateur est représentée en sac de mots puis le sac de mots qui représente l'utilisateur est la moyenne des sacs de mots de ses publications. Chaque valeur du sac de mots est considérée comme une caractéristique ; ainsi, nous les avons ajoutées aux vecteurs utilisateurs. La valeur d'une caractéristique, pour un uni-gramme donné u et pour une publication, est calculé par la formule suivante :

$$Fréquence\_uni - gramme(u) = \frac{Nombre\ de\ u}{Nombre\ total\ de\ mots} \tag{3.1}$$

TABLEAU 3.1 – Les 18 uni-grammes sélectionnés

im, like, dont, people, know, time, ive, even, much, feel, going, something, someone, day, things, life, though, help

3.3.1.2 Style de langue

Dans cette section, nous allons présenter les caractéristiques qui décrivent le style de langue ou le style d'écriture adopté par un utilisateur.

Fréquence des adjectifs, verbes, noms et adverbes : Choudhury *et al.* [Choudhury 2013] ont montré que les écrits des personnes suicidaires sont caractérisés par une utilisation accrue des verbes et des adverbes, et par une utilisation moindre des noms. Nous avons donc pris en compte ces résultats et les avons utilisés comme caractéristiques de détection de la dépression même si une personne dépressive n'est pas forcément suicidaire bien que l'Organisation Mondiale de la Santé ait montré que la dépression est l'une des causes importantes de suicide dans le monde [Organization 2017].

Nous obtenons ainsi un vecteur, pour chaque utilisateur, avec quatre valeurs représentant respectivement la fréquence normalisée des adjectifs, verbes, noms et adverbes. La fréquence normalisée d'une étiquette e (e = adjectif ou verbe ou nom ou adverbe) est donnée par la formule suivante :

$$Fréquence(e) = \frac{Nombre\ de\ e}{Nombre\ total\ de\ mots} \tag{3.2}$$

Fréquence de la Négation* : Souffrant de dépression, les gens expriment un sentiment beaucoup plus négatif dans leurs écrits. Nous avons donc décidé d'établir cette caractéristique. Ici, nous avons calculé la fréquence normalisée des mots négatifs les plus fréquemment utilisés (voir tableau 3.2).

$$Fréquence\_négation = \frac{Nombre\ de\ mots\ négatifs}{Nombre\ total\ de\ mots} \tag{3.3}$$

TABLEAU 3.2 – Liste des mots négatifs utilisés

aren, couldn, didn, doesn, hadn, hasn, haven, isn, mightn, mustn, needn, shan, shouldn, wasn, wouldn', no, not, didn't, didnot, don't, isn't, doesn't, can't, cannot, nobody, never, neither, either, nor, nothing, nowhere, none

Fréquence des ponctuations spéciales : Quand une personne souffre de dépression, ses écrits reflètent fidèlement son humeur. Dans ce cas, les signes de ponctuation spéciaux, tels que les points d'interrogation ou d'exclamation, tendent à inciter les utilisateurs à exprimer leurs doutes, leurs surprises, etc. vers une cible [Xue 2014].

Pour calculer cette caractéristique, il faut calculer la fréquence normalisée des points d'exclamation, points d'interrogation et/ou de toute combinaison de ces deux ponctuations (telles que !!!!, ???!!!, etc.).

$$Fréquence\_ponctuations = \frac{Nombre\ de\ ponctuations\ (?\ et\ !)}{Nombre\ total\ de\ ponctuations} \tag{3.4}$$

Fréquence des émoticônes : Les émoticônes sont une autre façon d'exprimer les ressentis pour les personnes. Xinyu Wang *et al.* [Wang 2013] ont montré que la prise en compte des émoticônes est importante dans la détection de la dépression.

Le calcul de la fréquence normalisée des émoticônes est basé sur un ensemble d'émoticônes glané soigneusement à partir de plusieurs plates-formes sociales par la doctorante Jiren Karoui.

$$Fréquence\_émoticônes = \frac{Nombre\ d'\ émoticônes}{Nombre\ total\ de\ mots} \tag{3.5}$$

Fréquence des mots en majuscule* : Cette caractéristique vient du fait que les personnes dépressives sont plus susceptibles de mettre l'accent sur la cible qu'elles mentionnent. L'utilisation de mots en capitales est une façon d'exprimer cette accentuation comme dans "I'm the UNLUCKIEST man in the world!". La valeur de cette caractéristique est la fréquence normalisée des mots en majuscule.

$$Fréquence\_mots\_majuscule = \frac{Nombre\ de\ mots\ en\ majuscule}{Nombre\ total\ de\ mots} \tag{3.6}$$

3.3.1.3 Comportement de l'utilisateur

Dans cette section, nous allons décrire toutes les caractéristiques qui décrivent le comportement de l'utilisateur. Plus précisément, nous allons voir le comportement de l'utilisateur lors de la rédaction de ses publications et ses heures de publications [Choudhury 2013].

Avant de présenter les caractéristiques suivantes, il est nécessaire de distinguer une publication postée par l'utilisateur lui-même (appelée "post") et une publication postée pour répondre aux publications des autres utilisateurs (appelée "commentaires").

Les 5 caractéristiques suivantes sont très spécifiques au format des jeux de données de la tâche eRisk [Losada 2018, Losada 2017] où les données ont été divisées en 10 partitions. Chaque partition contient 10 % des publications de chaque utilisateur (pour plus de détails, voir section 3.4). Pour la tâche eRisk, les partitions représentent la notion de temps, c'est-à-dire que la mise à disposition des données se fait partition par partition.

Dans ce qui suit, NTP désigne le nombre total de partitions.

Nombre moyen de posts : pour calculer la valeur de cette caractéristique, nous comptons d'abord les posts dans chaque partition, puis nous le divisons par le nombre total de partitions à disposition. Si le jeu de données n'est pas divisé en 10 partitions, cette caractéristique reviendrait juste à compter le nombre de posts de l'utilisateur.

$$Nombre\_moyen\_posts = \frac{\sum_{i=1}^{NTP} NPP_i}{NTP} \tag{3.7}$$

où NPP_i est le nombre de posts dans la partition i.

Nombre moyen de mots par posts : nous calculons le nombre moyen de mots par posts pour chaque partition et la moyenne de ces nombres sur toutes les partitions à disposition est la valeur de cette caractéristique. Si le jeu de données n'est pas divisé

en partitions, cette caractéristique est juste le nombre moyen de mots dans les posts de l'utilisateur.

$$Nombre\_moyen\_mot\_par\_posts = \frac{\sum_{i=1}^{NTP} \frac{NMPP_i}{NPP_i}}{NTP} \tag{3.8}$$

où $NMPP_i$ est le nombre total de mots dans les posts dans la partition i et NPP_i est le nombre de posts dans la partition i.

Nombre minimum de posts : nous comptons le nombre de posts dans chaque partition et le minimum de ces nombres est la valeur de cette caractéristique.

$$Nombre\_minimum\_posts = \min_{i=1}^{NTP}(NPP_i) \tag{3.9}$$

où NPP_i est le nombre de posts dans la partition i.

Nombre moyen de commentaires : cette caractéristique se calcule de la même manière que le nombre moyen de posts, mais pour les commentaires.

$$Nombre\_moyen\_commentaires = \frac{\sum_{i=1}^{NTP} NCP_i}{NTP} \tag{3.10}$$

où NCP_i est le nombre de commentaires dans la partition i.

Nombre moyen de mots par commentaires : cette caractéristique se calcule de la même manière que le nombre moyen de mots par posts, mais pour les commentaires.

$$Nombre\_moyen\_mot\_par\_commentaires = \frac{\sum_{i=1}^{NTP} \frac{NMCP_i}{NCP_i}}{NTP} \tag{3.11}$$

où $NMCP_i$ est le nombre total de mots dans les commentaires dans la partition i et NPP_i est le nombre de commentaires dans la partition i.

Heures de publications* : d'après Chouldhury *et al.*[Choudhury 2013], les habitudes de sommeil des utilisateurs dépressifs peuvent être irrégulières. Les utilisateurs dépressifs ont tendance à poster des publications tard dans la nuit. Nous avons donc pris en compte ces résultats pour capturer le ratio de publications en fonction des périodes de la journée. Pour cela, nous avons divisé une journée en 4 segments : "matin" (7am-12am), "après-midi" (12am-18pm), "nuit" (18pm-00pm), "nuit tardive" (00pm-7am). Comme les utilisateurs dépressifs auraient tendance à publier tard dans la nuit, nous ne considérons que le dernier segment. Le nombre de publications postées dans ce dernier segment est divisé par le nombre total de publications de l'utilisateur afin de normaliser les résultats.

$$Heures\_publications = \frac{Nombre\ de\ publications\ publiées\ dans\ nuit\ tardive}{Nombre\ total\ de\ publications} \tag{3.12}$$

Saison de publications : pour calculer ces caractéristiques, nous avons divisé les 12 mois de l'année en 4 saisons [5] ; chaque saison correspond à une caractéristique. La valeur de chaque caractéristique est égale au nombre de publications de l'utilisateur publiées dans la saison correspondante divisé par le nombre total de publications de l'utilisateur.

$$Saision_i = \frac{Nombre\ de\ publications\ publiées\ dans\ la\ saison\ i}{Nombre\ total\ de\ publications} \tag{3.13}$$

où $i \in \{1, 2, 3, 4\}$.

3.3.1.4 Préoccupation personnelle

Dans cette section, nous allons décrire les caractéristiques qui indiquent la préoccupation des utilisateurs pour eux-mêmes. Nous mesurons l'utilisation importante des pronoms personnels qui se réfèrent à l'auteur lui-même ou la sur-généralisation.
Fréquence des pronoms à la première personne : comme une personne dépressive a plus tendance à se préoccuper d'elle-même, nous nous attendons à une présence significative de pronoms à la première personne dans ses textes.

Nous avons calculé la fréquence normalisée de chaque pronom p (voir le tableau 3.3). Chaque pronom p correspond à une caractéristique. À part ces pronoms, nous avons aussi calculé la fréquence du pronom "I" quand il est sujet du verbe "be", plus précisément la fréquence du mot *I'm* [Rude 2004]. Nous avons aussi calculé la somme des fréquences des pronoms, mentionné précédemment (avec le mot *I'm*), afin de mieux capturer l'auto-référence des utilisateurs dépressifs.

$$Fréquence\_total\_pronoms = \left(\sum \frac{Nombre\ de\ p}{Nombre\ total\ de\ mots}\right) + \frac{Nombre\ de\ "I"m"}{Nombre\ total\ de\ mots} \tag{3.14}$$

TABLEAU 3.3 – Liste des pronoms à la première personne

I, me, myself, mine, my

Fréquence du pronom "I" dans un contexte subjectif* : au lieu de juste compter tous les "I" figurant dans les textes, nous examinons maintenant la présence du pronom

5. saison 1 : Décembre, Janvier, et Février ; saison 2 : Mars, Avril, et Mai ; etc.

«I» dans des contextes subjectifs. Pour cela, nous nous concentrons particulièrement sur tous les "I" suivis d'un adjectif, car cette catégorie grammaticale est souvent utilisée pour véhiculer un sens subjectif concernant l'utilisateur et est généralement utilisée pour exprimer des sentiments ou des émotions concernant leur cible. Le but ici est de capturer la fréquence à laquelle un utilisateur exprime des sentiments ou des émotions quand il parle de lui-même. Pour extraire ce trait, nous nous appuyons sur des patrons lexico-syntaxiques spécifiques tels que : `I 'm NEG ADJ` (exemple, *I'm not attractive*), and `I 'm ADV ADJ` (exemple, *I'm very nervous*).

$$Fréquence\_I\_contexte\_subjectif = \frac{Nombre\ de\ I\ dans\ un\ contexte\ subjectif}{Nombre\ total\ de\ mots} \tag{3.15}$$

Sur-généralisation ou Quantificateurs d'intensité : Mowery *et al.* [Mowery 2016] ont remarqué qu'une personne dépressive est encline à sur-généraliser ses propos. Par exemple, au lieu de critiquer quelqu'un qui l'a blessé(e) auparavant, la personne dépressive dira probablement quelque chose comme "tous/toutes les hommes/femmes sont méchants/méchantes".

Nous avons calculé la valeur de cette caractéristique en sommant la fréquence normalisée de chaque quantificateur q présenté dans le tableau 3.4. À ces quantificateurs, nous avons ajouté des superlatifs comme *worst*.

$$Sur\_généralisation = \sum \frac{Nombre\ de\ q}{Nombre\ total\ de\ mots} \tag{3.16}$$

TABLEAU 3.4 – Liste des quantificateurs

all, everything, everyone, everywhere, everytime, anyone, anything, anywhere, anytime, nothing, most, more, nobody, no one, anymore, never and ever

3.3.1.5 Réminiscence

Cette section rapporte les caractéristiques qui font allusion au passé. En effet, selon Mowery *et al.* [Mowery 2016], les utilisateurs dépressifs ont souvent tendance à se référer au passé.

Expressions temporelles faisant référence au passé : pour calculer cette caractéristique, nous avons utilisé une liste de marqueurs en anglais, faisant référence au passé, fréquemment utilisés (voir le tableau 3.5).

TABLEAU 3.5 – Liste de marqueurs faisant référence au passé

yesterday, last, before, ago, past, back, earlier, later, nostalgia

Nous avons calculé la valeur de cette caractéristique en sommant les fréquences normalisées des marqueurs m dans le tableau 3.5.

$$Expressions\_temporelles = \sum \frac{Nombre\ de\ m}{Nombre\ total\ de\ mots} \tag{3.17}$$

Verbes conjugués au passé : nous avons additionné les occurrences des verbes conjugués au passé puis le total est divisé par le nombre total de verbes dans les textes de l'utilisateur.

$$Fréquence\_verbe\_passé = \frac{Nombre\ de\ verbes\ conjugués\ au\ passé}{Nombre\ total\ de\ mots} \tag{3.18}$$

Fréquence du passé : c'est la combinaison des deux caractéristiques précédentes. Plus précisément, il s'agit de la fréquence normalisée des verbes conjugués au passé et des marqueurs faisant référence au passé.

$$Fréquence\_du\_passé = Fréquence\_verbe\_passé + Expressions\_temporelles \tag{3.19}$$

Auxiliaires "être" conjugués au passé (was, were) : la valeur de cette caractéristique est la fréquence normalisée des auxiliaires "was" et "were".

$$Fréquence\_auxiliaires = \frac{Nombre\ de\ "was"\ et\ "were"}{Nombre\ total\ de\ mots} \tag{3.20}$$

3.3.1.6 Marqueurs de la dépression

Dans cette section, nous allons présenter les caractéristiques plus spécifiques à la dépression comme les noms des médicaments, etc.

Symptômes de la dépression et des noms d'antidépresseurs : les utilisateurs dépressifs mentionnent souvent des symptômes de la dépression et des noms d'antidépresseurs dans leurs textes. Nous avons utilisé comme ressource, une liste L de symptômes de la dépression et de noms d'antidépresseurs obtenus de [Choudhury 2013] et sur Wikipédia. La valeur de cette caractéristique est la fréquence normalisée des unigrammes dans la liste.

$$Fréquence\_Symptômes = \frac{Nombre\ d'\ uni\text{-}grammes\ de\ la\ liste\ L}{Nombre\ total\ de\ mots} \tag{3.21}$$

Noms chimiques et marques des antidépresseurs : cette caractéristique est semblable à la précédente sauf qu'elle capture des détails plus précis (ou techniques) sur les antidépresseurs. Nous avons utilisé une liste obtenue de WebMD [6].

$$Fréquence\_noms\_chimiques = \frac{Nombre\ d'\ uni\text{-}grammes\ de\ la\ liste\ WebMD}{Nombre\ total\ de\ mots} \tag{3.22}$$

Fréquence du mot "depress"* : comme notre objectif est de détecter la dépression, il est naturellement utile de saisir directement le mot-clé "depress" et ses variantes morphologiques comme *depressing, depressed, depression, depressive, etc.*. La valeur de cette caractéristique est la fréquence normalisée du mot "depress" et de ses variantes.

$$Fréquence\_depress = \frac{Nombre\ du\ mot\ "depress"\ et\ de\ ses\ variantes}{Nombre\ total\ de\ mots} \tag{3.23}$$

Cependant, il faut noter que si cette caractéristique est vraiment efficace pour la détection de la dépression, elle n'est ni généralisable ni extensible à une tâche de détection automatique sauf si cette dernière est très liée à la dépression.

Fréquence des mots liés au sommeil* : nous avons remarqué que les utilisateurs dépressifs ont tendance à parler de leur sommeil dans leurs publications en utilisant des mots comme "sleep". Cela est peut-être dû aux habitudes de sommeil des personnes dépressives, souvent influencées négativement par la dépression. La valeur de cette caractéristique est la fréquence normalisée du mot "sleep".

$$Fréquence\_sommeil = \frac{Nombre\ du\ mot\ "sommeil"}{Nombre\ total\ de\ mots} \tag{3.24}$$

Fréquences des 25 tri-grammes et des 25 5-grammes : les utilisateurs dépressifs ont tendance à utiliser des groupes de mots comme "want to die" (tri-grammes) ou "To take my own life" (5-grammes), etc. Nous avons directement utilisé les 25 tri-grammes et 25 5-grammes, qui étaient les plus fréquents exprimant une pensée potentiellement suicidaire, extraits d'une collection de Tweets par Colombo *et al.* [Colombo 2016].

6. `https://www.streetdirectory.com/travel_guide/15675/writing/how_to_choose_the_best_readability_formula_for_your_document.html`, consulté le 25 Février 2018.

Notons que nous avons ici deux caractéristiques, la fréquence normalisée des trigrammes et la fréquence normalisée des 5-grammes.

$$Fréquence\_(3\ ou\ 5)\text{-}grammes = \frac{Nombre\ des\ (3\ ou\ 5)\text{-}grammes\ de\ Colombo}{Nombre\ total\ de\ (3\ ou\ 5)\text{-}grammes} \tag{3.25}$$

3.3.1.7 Sentiment et émotion

Dans cette section, nous allons décrire les caractéristiques qui permettent de quantifier les sentiments et émotions des utilisateurs.

Analyse des sentiments : ici nous avons deux caractéristiques : la fréquence de sentiments négatifs et la fréquence de sentiments positifs. Les personnes dépressives ont tendance à être plus subjectives par rapport à ce qu'elles mentionnent ou écrivent, c'est pourquoi nous avons supposé qu'il est utile de tracer la polarité de leurs textes. La fréquence des sentiments négatifs (respectivement sentiments positifs) est simplement la fréquence normalisée des mots négatifs (respectivement des mots positifs).

Pour obtenir la liste des mots positifs et négatifs, nous avons utilisé le "NRC-Sentiment-Emotion-Lexicons [7] [8]" [Mohammad 2013].

$$Fréquence\_(positive\ ou\ négative) = \frac{Nombre\ des\ mots\ de\ la\ liste\ (positive\ ou\ négative)}{Nombre\ total\ de\ mots} \tag{3.26}$$

Fréquence des émotions : pour calculer la fréquence des émotions, nous avons calculé la fréquence normalisée des mots associés à des émotions étroitement liées à la dépression.

De façon similaire à la création des listes de mots négatifs et positifs, nous avons aussi créé une liste de mots exprimant des émotions étroitement liées à la dépression. Nous avons aussi utilisé le lexique "NRC-Sentiment-Emotion-Lexicons" [Mohammad 2013] qui contient huit types d'émotion. Parmi ces types d'émotion, nous en avons sélectionné cinq que nous considérons comme étroitement liées à la dépression : anger, fear, surprise, sadness et disgust. Tous les mots, qui sont donc associés à

7. `http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm`, consulté le 08 Avril 2019

8. C'est un lexique contenant une liste de mots en anglais et leurs associations aux deux sentiments (positif et négatif) ainsi qu'à huit types d'émotions (anger, fear, anticipation, trust, surprise, sadness, joy, et disgust).

ces cinq émotions, sont ajoutés dans notre liste.

$$Fréquence\_émotions = \frac{Nombre\ des\ mots\ de\ la\ liste\ des\ émotions}{Nombre\ total\ de\ mots} \tag{3.27}$$

3.3.1.8 Autres

Lisibilité d'un texte : l'objectif ici est de mesurer la complexité de l'écriture d'une publication d'un utilisateur. Nous avons défini 4 caractéristiques qui sont 4 mesures de lisibilité d'un texte : Gunning Fog Index (FOG), Flesch Reading Ease (FRE), Linsear Write Formula (LWR), New Dale-Chall Readability (DCR). FOG estime les années d'études dont une personne a besoin pour comprendre le texte à la première lecture, FRE mesure la difficulté de la compréhension d'un texte, LWR a été développé pour calculer la lisibilité des manuels techniques de l'US Air Force, et enfin DCR mesure la difficulté de compréhension que les personnes rencontrent lors de la lecture d'un texte. Pour un utilisateur donné, chaque caractéristique est la moyenne des mesures (la mesure correspondant à la caractéristique) sur toutes les publications de l'utilisateur.

Catégorie textuelle avec Empath* : l'idée ici est d'analyser la publication à travers des catégories lexicales. Pour cela, nous avons utilisé une librairie de python appelée empath [9] qui correspond à un outil d'analyse de textes à travers des catégories lexicales. Par défaut, empath comporte 194 catégories lexicales (voir Annexe B pour la liste des catégories) et chaque catégorie, notée cat, est considérée comme une caractéristique dans notre travail. Pour un texte donné, empath renvoie un score pour chaque catégorie, ces scores sont une estimation de l'appartenance du texte à ces catégories lexicales. Dans notre travail, la valeur d'une catégorie donnée cat, c'est-à-dire d'une caractéristique, est le score renvoyé par empath pour cette catégorie sur un texte qui est la concaténation des publications de l'utilisateur.

$$Catégorie\_textuelle(cat) = empath_{cat}(publications) \tag{3.28}$$

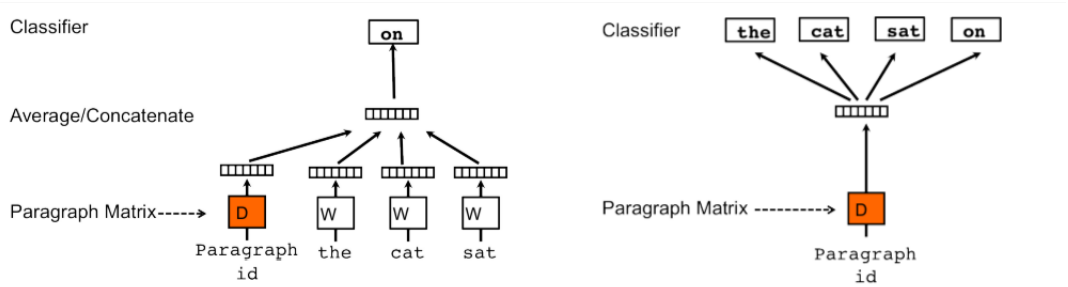
Au total, nous avons eu 256 caractéristiques donc un vecteur utilisateur est de dimension 256.

9. `https://github.com/Ejhfast/empath-client`, consulté le 5 Septembre 2018

3.3.2 Représentation de l'utilisateur par la méthode de plongement de phrases

Dans cette section, nous indiquons comment représenter les utilisateurs par des vecteurs en utilisant le plongement de phrases. Comme dans la section 3.3.1 où un utilisateur est représenté par un vecteur composé de caractéristiques calculées à partir de plusieurs publications, ici aussi l'utilisateur est représenté par un seul vecteur qui représente ses publications. Pour obtenir ce vecteur, nous avons procédé comme suit : premièrement, chaque publication est représentée par un vecteur obtenu par le plongement de phrases puis la moyenne de ces vecteurs est réalisée pour obtenir le vecteur final représentant l'utilisateur.

Dans ce travail, nous allons utiliser le plongement de phrases Doc2Vec [Le 2014] qui est basé sur le plongement de mots Word2Vec [Mikolov 2013b] présenté dans le Chapitre 2. Le concept est simple : ajouter un autre vecteur (id du paragraphe) au modèle Word2vec (cf. la figure 3.2). L'objectif de Doc2vec est de créer une représentation numérique d'un document/texte, quelle que soit sa longueur. Doc2vec peut très bien fonctionner quand il est entraîné sur un petit corpus par rapport à Word2vec [Trotzek 2017].



(a) Mémoire distribuée *(b) Sac de mots distribué*

FIGURE 3.2 – Architecture des deux modèles de Doc2vec [Le 2014]

Dans notre travail, nous avons entraîné deux modèles de Doc2vec sur les données d'entraînement en utilisant la bibliothèque gensim de Python :

1) **Modèle de mémoire distribuée** : c'est une petite extension du modèle CBOW de Word2vec, mais au lieu d'utiliser seulement des mots pour prédire le mot suivant, ce modèle de Doc2vec ajoute un autre vecteur caractéristique en entrée qui est unique au document (id du paragraphe sur la figure 3.2 (a)). Durant l'entraînement, les vecteurs de mots W et le vecteur de document D sont entraînés

(c'est-à-dire changent de poids) et à la fin, le vecteur de document D contient une représentation numérique du document. Chaque vecteur de document est affecté à un seul document tandis que les vecteurs de mots sont partagés entre tous les documents.

2) **Modèle de sac de mots distribué** : ce modèle, similaire au modèle skip-gram de Word2vec, utilise le vecteur de paragraphes pour classer des mots entiers dans le document/texte (cf. figure 3.2 (b)). Au lieu de prédire le mot suivant, il utilise le vecteur de document pour classer des mots entiers dans le document. Plus précisément, durant la phase d'entraînement, un classifieur est entraîné pour décider si des mots appartiennent au document (donné en entrée) ou non. Ce modèle consomme moins de mémoire, car il n'est pas nécessaire de sauvegarder les vecteurs de mots comme dans le modèle de mémoire distribuée.

Dans leur article, Mikolov *et al.* [Le 2014] recommandent d'utiliser une combinaison des deux modèles, bien que le modèle de mémoire distribuée obtienne généralement des résultats comparables à ceux de l'état de l'art.

Dans notre travail, pour représenter un texte, nous allons aussi concaténer les deux vecteurs représentant le texte, obtenus par les deux modèles, formant ainsi un seul vecteur de dimension 200.

3.4 Tâche eRisk et jeu de données

Dans cette section, nous décrivons d'abord les jeux de données des deux éditions de la tâche eRisk (2017 et 2018) puisque nous évaluons notre approche sur ces données. Ensuite, nous décrivons la tâche eRisk pour une meilleure compréhension du problème que nous traitons.

3.4.1 Jeu de données eRisk 2017 & 2018

Nous présentons ici les jeux de données eRisk 2017 & 2018 qui ont été utilisés pour la tâche eRisk.

Les données de la tâche eRisk ont pour objectif de prendre en compte l'évolution du langage utilisé par les personnes dépressives en analysant une longue séquence chronologique de publications. Le facteur temps est considéré comme fondamental dans la construction d'une telle ressource dans la mesure où une prise en charge de la maladie dès ses débuts est préférable [Losada 2016]. Ainsi, l'objectif principal de la

tâche eRisk est de détecter au plus tôt les indices de dépression [10][Losada 2016].

Les jeux de données sont composés de publications en anglais issues de la plateforme de réseaux sociaux Reddit [11]. Le contenu de Reddit est organisé par thèmes appelés "subreddits". Les utilisateurs peuvent publier des posts ou répondre/commenter les posts et commentaires des autres utilisateurs. Les posts et commentaires sont appelés des publications. Les publications sont représentées par l'identifiant de l'utilisateur, l'heure de publication et le contenu textuel.

Pour construire les jeux de données eRisk, Losada et Crestani ont collecté un nombre maximal de publications dans tous les "subreddits" pour chaque utilisateur. Les utilisateurs ayant moins de 10 publications ont été exclus. Dans cet ensemble de données, les utilisateurs ont été annotés comme *dépressifs* ou *non dépressifs*. Pour considérer un utilisateur comme dépressif, il/elle doit avoir une publication contenant une auto-déclaration de dépression, telle que "J'ai été diagnostiqué dépressif". Les organisateurs ont vérifié manuellement le texte et annoté l'utilisateur correspondant comme dépressif. Ce texte contenant l'auto-déclaration a ensuite été supprimé de l'ensemble de données afin de rendre la détection non triviale (les autres publications de l'utilisateur sont en revanche conservées). Les utilisateurs dont les publications ne contiennent aucune auto-déclaration de dépression ont été considérés comme non dépressifs. Le jeu de données est décrit en détail dans [Losada 2016].

Pour chaque utilisateur, la collection de textes est une séquence de publications triées par ordre chronologique. Les jeux de données ont ensuite été divisés en 10 partitions (partitions), la partition 1 contient 10 % des publications (les plus anciennes) de chaque utilisateur, la partition 2 contient les 10 % de publications suivantes, etc. La figure 3.3 montre un exemple de contenu d'une publication d'un utilisateur annoté comme "dépressif".

> I was feeling much better, myself harm stopped/suicidal thoughts stopped, I became more social, I could focus on school/get things done. Then my mom noticed that I wasn't eating as much as I used to, and decided to do some research about the medication I was on. She made me stop taking it immediately afterwards...

FIGURE 3.3 – Exemple de texte d'un utilisateur annoté comme dépressif

Chacune des 10 partitions d'un jeu de données contient des fichiers XML (un fichier

10. http://early.irlab.org/
11. https://www.reddit.com/

par utilisateur) qui stockent l'identifiant de l'utilisateur et ses publications. Chaque publication de l'utilisateur contient le titre, l'heure et le contenu textuel de la publication. Si le titre est vide, la publication est considérée comme un commentaire, sinon il s'agit d'un post. La figure 3.4 présente un exemple d'un post et d'un commentaire de la collection eRisk.

```
<WRITING>
        <TITLE> I need help finding data on US presidential elections </TITLE>
        <DATE> 2014-04-09 04:34:40 </DATE>
        <INFO> reddit post </INFO>
        <TEXT> I was wondering if you could help me find some information.
I'm looking for statistics on the past 20 US presidential elections, such
has how people voted based on their religion, ethnicity, education, etc.
For example, "38% of catholics voted for Clinton." I've been searching all
over for this data, and any help on finding it would be majorally appreciated. </TEXT>
</WRITING>
```

```
<WRITING>
        <TITLE>   </TITLE>
        <DATE> 2014-04-09 17:03:41 </DATE>
        <INFO> reddit post </INFO>
        <TEXT> Sorry to bother you again, I was wondering if you could
help me with another thing. Do you know where I could find data on the
same topic, but this time, what % turned up for the election? For example,
"38% of Catholics said they voted in 1984." If you can help, it would
be a lot of help :) thanks afaib </TEXT>
</WRITING>
```

FIGURE 3.4 – Exemple d'un post (en haut) et un commentaire (en bas) de la collection eRisk 2017.

Un jeu de données est divisé en deux : des données d'entraînement et des données de test.

Le tableau 3.6 décrit la répartition pour les jeu eRisk 2017 et 2018 sur la dépression. Concernant le jeu de données de la première édition d'eRisk (2017), les données d'entraînement (respectivement données de test) contiennent 83 (52) utilisateurs annotés "dépressifs" et 403 (349) utilisateurs annotés "non dépressifs". Nous pouvons constater que le ratio entre utilisateurs dépressifs et non dépressifs n'est pas équilibré dans le jeu d'apprentissage (0.21) et celui de test (0.15). Ce déséquilibre de classes rend la classification des utilisateurs plus difficile.

Concernant le jeu de données de la deuxième édition d'eRisk ; il comprend 214 utilisateurs annotés dépressifs et un groupe de contrôle de 1 493 utilisateurs. Les données d'entraînement (respectivement données de test) contiennent 135 (79) utilisateurs annotés "dépressifs" et 752 (741) utilisateurs annotés "non dépressifs". Nous pouvons constater que, comme pour le jeu de données de eRisk 2017, le ratio entre utilisateurs dépressifs et non dépressifs n'est pas équilibré dans le jeu d'apprentissage (0.18) et celui de test (0.11). Notons que les données d'entraînement de ce jeu de données de eRisk 2018 sont composées des données d'entraînement et de test du jeu de données de eRisk

2017.

TABLEAU 3.6 – Distribution des données d'entraînement et de test des collections eRisk (2017 et 2018) sur la dépression.

Nombre de		*Entraînement*		*Test*	
		Dépressif	Non dépressif	Dépressif	Non dépressif
2017	Utilisateurs	83	403	52	349
	Posts	4,911	91,381	1,928	65,735
	Commentaires	25,940	172,791	16,778	151,930
2018	Utilisateurs	135	752	79	741
	Posts	6,839	157,116	7,672	169,930
	Commentaires	42,718	324,721	32,993	333,852

Le jeu de données eRisk 2018 comprend également des annotations sur l'anorexie. Le tableau 3.7 présente la distribution du jeu de données. Les données d'entraînement (respectivement données de test) contiennent 20 (41) utilisateurs annotés "anorexiques" et 132 (279) utilisateurs annotés "non anorexiques".

TABLEAU 3.7 – Distribution des données d'entraînement et de test de la collection eRisk 2018 sur l'anorexie.

Nombre de	Entraînement		Test	
	Anorexique	Non Anorexique	Anorexique	Non Anorexique
Utilisateurs	20	132	41	279
Posts	2,009	21,624	2,096	35,781
Commentaires	5,443	55,758	15,326	115,304

3.4.2 La tâche eRisk

L'objectif de la tâche eRisk est de détecter le plus tôt possible les signes de la dépression (ou d'anorexie) dans les textes des utilisateurs. Pour mesurer ce facteur de temporalité, comme indiqué précédemment, les données ont été divisées en 10 partitions selon la chronologie des publications. Durant la phase d'entraînement, les 10 partitions du jeu d'entraînement sont données en même temps aux utilisateurs pour

créer leurs modèles. Durant la phase de test de la tâche, une partition est donnée aux participants chaque semaine, en commençant par les publications les plus anciennes des utilisateurs. Les participants ont la possibilité de classer un utilisateur comme dépressif ou non dépressif ou de retarder la décision afin d'utiliser des données additionnelles de la semaine suivante. Les décisions émises durant la semaine sont définitives et ne peuvent plus être changées plus tard. La dernière semaine, une décision doit être prise pour chaque utilisateur. Pour évaluer les systèmes, les organisateurs ont utilisé une mesure appelée Early Risk Detection Error (ERDE) (voir section 3.5 pour plus de détails sur la mesure), une mesure qui prend en compte la véracité des décisions prises ainsi que le nombre de publications utilisées pour prendre ces décisions. Plus les systèmes des participants utilisent de partitions (c'est-à-dire de données) pour prendre les décisions, plus ils sont pénalisés.

Dans ce travail, nous entraînons d'abord des classifieurs sans tenir compte de la dimension temps (partitions), c'est-à-dire que nous utilisons toutes les données disponibles dans le corpus d'entraînement (les 10 partitions). Ensuite, nous utilisons ces classifieurs entraînés pour la détection au plus tôt sur les données de test. Pour s'assurer d'avoir des résultats comparables à ceux des participants à la tâche eRisk, nous avons suivi le même protocole que lors de la phase de test de la participation à la tâche, c'est-à-dire en procédant partition par partition. Avec les données de la première partition, nos classifieurs entraînés prédisent si un utilisateur est dépressif, sinon nous utilisons la deuxième partition en plus des données de la première partition) et nous renouvelons la prédiction. Ce processus est répété jusqu'à la dernière partition. Après la dernière partition, tous les utilisateurs qui n'ont pas encore été classés comme dépressifs sont classés comme non dépressifs. Pour une partition donnée, les données utilisées pour la classification sont le cumul des données dans cette dite partition et de celles des partitions précédentes s'il y en a.

3.5 Expérimentations et Résultats

Dans cette section, nous présentons les mesures d'évaluation utilisées pour la tâche eRisk, ensuite nous décrivons les classifieurs que nous avons utilisés ainsi que les processus d'entraînement, et enfin les résultats que nous avons obtenus.

3.5.1 Mesures d'évaluation

Comme le problème que nous traitons est de détecter au plus tôt la dépression, une nouvelle mesure a été définie. Cette nouvelle mesure prend en compte l'exactitude

des décisions prises par le système, mais aussi le temps qu'il prend pour émettre cette décision. Cette mesure est appelée ERDE et a été définie par Losada *et al.* [Losada 2016] comme suit :

$$ERDE_o(d,k) = \begin{cases} c_{fp} & si\ d\ =\ positif\ et\ vérité\ terrain\ =\ négatif \\ c_{fn} & si\ d\ =\ négatif\ et\ vérité\ terrain\ =\ positif \\ lc_o(k)\cdot c_{tp} & si\ d\ =\ positif\ et\ vérité\ terrain\ =\ positif \\ 0 & si\ d\ =\ négatif\ et\ vérité\ terrain\ =\ négatif \end{cases} \tag{3.29}$$

Où :

- d est la décision binaire prise par le système avec le délai k ;
- $c_{fn} = c_{tp} = 1$;
- c_{fp} = proportion de cas positifs dans les données de test
- $lc_o(k) = 1 - \frac{1}{1+e^{k-o}}$
- k est le délai pris par le système pour prendre sa décision, c'est-à-dire le nombre de textes utilisés ;
- o est un paramètre et est égal à 5 pour ERDE_5 et est égal à 50 pour ERDE_{50}.

Notons que la valeur ERDE d'un système est la moyenne des ERDE calculés pour chaque utilisateur avec la formule 3.29. Aussi, plus la valeur de la mesure ERDE est petite, meilleur est le système.

Nous avons aussi considéré d'autres mesures standard utilisées pour évaluer les systèmes de classifications, qui sont : la précision (P), le rappel (R) et la f1-mesure (F).

$$P = \frac{Nombre\ d'utilisateurs\ correctement\ classés\ comme\ dépressifs}{Nombre\ total\ d'utilisateurs\ classés\ par\ le\ système\ comme\ dépressifs} \tag{3.30}$$

$$R = \frac{Nombre\ d'utilisateurs\ correctement\ classés\ comme\ dépressifs}{Nombre\ total\ d'utilisateurs\ réellement\ dépressifs} \tag{3.31}$$

$$F = 2 \times \frac{R \times P}{R + P} \tag{3.32}$$

3.5.2 Entraînement des modèles

Nous avons utilisé différents classifieurs utilisant un apprentissage supervisé tels que les forêts d'arbres de décision (RF), la régression logistique (RL), SVM et le réseau de neurones perceptron multicouche (MLP). Durant l'entraînement des modèles, nous avons utilisé toutes les données dans le corpus, c'est-à-dire les 10 partitions de données. Les vecteurs utilisateurs sont construits à partir de toutes ces données. Il y a deux types de vecteurs utilisateurs comme nous avons vu dans la section 3.3 : les vecteurs construits avec le plongement de phrases et ceux construits avec des caractéristiques. Pour tester les différents classifieurs sur ces deux types de vecteurs, nous avons utilisé la validation croisée avec 10 parties et les résultats sont présentés dans le tableau 3.8.

TABLEAU 3.8 – Résultats obtenus avec la validation croisée sur les jeux de données eRisk 2017 et 2018. Les meilleurs résultats sont en gras.

Classifieur	Plongement de phrases		Caractéristiques	
	2017	2018	2017	2018
RF	0.76	0.70	**0.78**	**0.76**
LR	**0.78**	**0.74**	0.59	0.60
SVM	0.72	0.69	0.77	0.75
MLP	0.76	0.74	0.75	0.73

Nous constatons que le meilleur modèle, sur les deux jeux de données, utilise le classifieur régression logistique (RL) quand les données sont transformées avec la méthode de plongement de phrases et utilise le classifieur forêts d'arbres de décision (RF) avec les caractéristiques. Dans la suite de ce manuscrit, quand nous parlons de modèles qui utilisent les forêts d'arbres de décision, cela signifie qu'ils sont entraînés sur les vecteurs caractéristiques et les modèles qui utilisent la régression logistique sont entraînés sur les vecteurs du plongement de phrases. Notons qu'à partir de maintenant, les modèles sont entraînés sur l'ensemble du corpus d'entraînement pour chaque jeu de données.

Nous avons aussi effectué une réduction des caractéristiques puisqu'au total nous en avons 256. Pour la sélection des caractéristiques, nous avons utilisé la méthode du "Chi-squared ranking". Sur le jeu de données de eRisk 2017, 124 caractéristiques sont sélectionnées dont 83 sont des caractéristiques que nous avons proposées. Sur le jeu de données de eRisk 2018, ce sont 154 caractéristiques qui sont retenues dont 109 font partie de nos propositions. Nous avons aussi constaté que 123 caractéristiques sont communes aux 124 et 154 caractéristiques sélectionnées, soit presque la totalité de

celles retenues pour le jeu de données de eRisk 2017. Ceci est tout à fait logique vu que les ensembles du jeu de données de eRisk 2017 (entraînement + test) forment l'ensemble d'entraînement du jeu de données de eRisk 2018. La liste de ces caractéristiques sélectionnées est présentée dans le tableau 5.1 (Annexe A). Nous reportons aussi les résultats des modèles entraînés avec ces caractéristiques filtrées et en utilisant le classifieur forêts d'arbres de décision.

3.5.3 Résultats

Dans cette section, nous présentons les résultats de notre approche sur les deux jeux de données de eRisk sur la détection de la dépression. Nous avons, pour cela, proposé cinq modèles qui sont :

- ModRF_256_traits : ce modèle est construit avec le classifieur RF en utilisant la représentation des utilisateurs avec les 256 caractéristiques.
- ModRF_traits_réduits : ce modèle est aussi construit avec le classifieur RF, mais cette fois-ci en utilisant les caractéristiques filtrées, c'est-à-dire 124 pour le jeu de données 2017 et 154 pour celui de 2018.
- ModLR : ce modèle est construit avec le classifieur RL en utilisant la représentation des utilisateurs avec le plongement de phrases.
- ModComb_moyenne : ce modèle est la combinaison des deux modèles : ModRF_traits_réduits et ModLR. Pour cela, la probabilité en sortie de ce modèle est la moyenne des probabilités en sortie des deux modèles qui le composent.
- ModComb_max : ce modèle est aussi une combinaison des deux modèles : ModRF_traits_réduits et ModLR. Ce modèle prend comme probabilité en sortie, le maximum des probabilités en sortie des deux modèles qui le composent.

Ces modèles produisent en sortie des probabilités où une probabilité mesure l'appartenance d'un utilisateur à la classe "dépression". C'est à l'aide de ces probabilités que nos modèles vont prendre des décisions à chaque partition de données, c'est-à-dire classer l'utilisateur comme dépressif ou attendre une nouvelle partition. Pour cela, nous avons défini des seuils de probabilité, plus précisément, nous en avons défini deux :

- seuil simple : un utilisateur est classé comme dépressif quand le modèle le prédit avec une probabilité supérieure à 0.5.
- seuil complexe : un utilisateur est classé comme dépressif si le modèle le prédit avec une probabilité supérieure à 0.55 et qu'il a au moins utilisé 20 publications de l'utilisateur pour calculer cette probabilité ; une probabilité supérieure à 0.7 et au moins 10 publications ; une probabilité supérieure à 0.5 et au moins 200 publications ; et enfin avec une probabilité supérieure à 0.9. Ce seuil est inspiré

du travail de Trotzek *et al.* [Trotzek 2017] que nous avons testé empiriquement sur le jeux de données d'entraînement de eRisk 2017 sur la dépression.

Résultats sur la détection de la dépression

Nous présentons dans ce qui suit les résultats de nos modèles pour la détection au plus tôt de la dépression sur les jeux de données eRisk 2017 et 2018.

Le tableau 3.9 présente les résultats des modèles en utilisant le seuil simple sur le jeu de données de eRisk 2017, alors que le tableau 3.10 présente les résultats utilisant le seuil complexe.

TABLEAU 3.9 – Résultats avec le seuil simple pour la détection au plus tôt de la dépression sur le jeu de données 2017. Les meilleurs résultats sont en gras.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_256_traits	12.88 %	10.37 %	0.59	0.65	0.54
ModRF_traits_réduits	**12.73** %	10.05 %	**0.64**	**0.69**	0.60
ModLR	14.50 %	9.35 %	0.43	0.29	0.81
ModComb_moyenne	13.33 %	**8.95** %	0.56	0.44	0.75
ModComb_max	14.49 %	9.10 %	0.45	0.30	**0.87**

TABLEAU 3.10 – Résultats avec le seuil complexe pour la détection au plus tôt de la dépression sur le jeu de données 2017. Les meilleurs résultats sont en gras.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_256_traits	**13.32** %	10.58 %	0.57	**0.69**	0.48
ModRF_traits_réduits	13.42 %	10.93 %	**0.60**	0.67	0.54
ModLR	14.12 %	8.78 %	0.49	0.36	0.77
ModComb_moyenne	13.96 %	9.61 %	0.56	0.51	0.62
ModComb_max	14.12 %	**8.54** %	0.52	0.38	**0.83**

En observant les deux tableaux (3.9 et 3.10), nous pouvons constater que ce sont les modèles utilisant les caractéristiques qui offrent les meilleurs résultats sur les mesures $ERDE_5$, f1-mesure (F) et précision (P). Ces modèles sont donc plus précis. Cela est peut-être dû au fait que les caractéristiques capturent mieux les signes de la dépression dans les textes. Le modèle basé sur le plongement de phrases, quant à lui, est meilleur sur les mesures $ERDE_{50}$ et rappel (R).

En combinant les modèles, nous observons une amélioration sur les deux mesures $ERDE_{50}$ et rappel (R) offrant ainsi les meilleurs résultats sur ces mesures. Une analyse plus détaillée devra être menée pour expliquer ce phénomène. Lorsque l'on considère les autres mesures par contre, les résultats des modèles combinés sont entre les résultats des modèles qui les composent, ce qui est peut-être plus logique. Nous pouvons aussi constater qu'en utilisant le seuil simple, la réduction des caractéristiques améliorent les résultats alors que c'est l'inverse en utilisant le seuil complexe (voir ModRF_256_traits et ModRF_traits_réduits).

Les deux tableaux suivants présentent les résultats des modèles sur le jeu de données eRisk 2018. Le tableau 3.11 pour le seuil simple et le tableau 3.12 pour le seuil complexe.

TABLEAU 3.11 – Résultats avec le seuil simple pour la détection au plus tôt de la dépression sur le jeu de données 2018. Les meilleurs résultats sont en gras.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_256_traits	9.55 %	7.05 %	0.57	0.67	0.49
ModRF_traits_réduits	9.62 %	6.92 %	**0.58**	**0.69**	0.51
ModLR	9.63 %	6.51 %	0.44	0.30	**0.81**
ModComb_moyenne	**8.94** %	**6.31** %	0.54	0.46	0.65
ModComb_max	9.63 %	6.51 %	0.44	0.30	**0.81**

TABLEAU 3.12 – Résultats avec le seuil complexe pour la détection au plus tôt de la dépression sur le jeu de données 2018. Les meilleurs résultats sont en gras.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_256_traits	9.79 %	7.96 %	0.49	0.70	0.38
ModRF_traits_réduits	9.79 %	7.84 %	0.52	**0.71**	0.41
ModLR	**9.52** %	**6.12** %	0.51	0.38	**0.80**
ModComb_moyenne	10.07 %	6.65 %	**0.56**	0.55	0.57
ModComb_max	**9.52** %	**6.12** %	0.51	0.38	**0.80**

Dans les deux tableaux (3.11 et 3.12), nous pouvons observer que les modèles utilisant les caractéristiques sont toujours les plus précis en offrant les meilleurs résultats sur les mesures précision (P) et f1-mesure (F). Mais contrairement aux résultats sur eRisk 2017, ils n'offrent plus le meilleur résultat sur la mesure $ERDE_5$. Le modèle basé sur le plongement de phrases, quant à lui, est toujours performant sur les me-

sures $ERDE_{50}$ et rappel (R), en offrant même les meilleurs résultats, ce qui n'était pas le cas sur eRisk 2017.

En combinant les modèles, nous observons aussi une amélioration sur deux mesures, mais cette fois-ci sur $ERDE_5$ et $ERDE_{50}$. Sur eRisk 2017, c'étaient les mesures $ERDE_{50}$ et rappel (R). Sur les autres mesures, les résultats sont soit égaux aux résultats d'un modèle composant les modèles combinés, soit entre les résultats des modèles qui les composent. Nous constatons aussi que la réduction des caractéristiques améliore les résultats quel que soit le type de seuil utilisé.

Nous avons comparé nos meilleurs modèles par rapport aux meilleurs résultats obtenus par les participants à la tâche eRisk, mais aussi par rapport aux meilleurs résultats dans la littérature. Cette comparaison est présentée dans le tableau 3.13 sur le jeu de données eRisk 2017 et dans le tableau 3.14 sur le jeu de données eRisk 2018.

Nous n'avons pas effectué des tests statistiques. Les résultats des autres modèles dans les deux tableaux (3.13 et 3.14) ont été directement tirés des articles qui les présentent. Nous n'avons pas eu en notre possession les codes pour obtenir des détails sur les résultats afin d'effectuer les tests.

TABLEAU 3.13 – Résultats de nos meilleurs modèles comparés à ceux de la littérature pour la détection au plus tôt de la dépression sur le jeu de données 2017.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_traits_réduits (seuil simple)	12.73 %	10.05 %	0.64	0.69	0.60
ModComb_max (seuil simple)	14.49 %	9.10 %	0.45	0.30	0.87
ModComb_max (seuil complexe)	14.12 %	8.54 %	0.52	0.38	0.83
FHDOB	12.70 %	10.39 %	0.55	0.69	0.46
FHDOA	12.82 %	9.69 %	0.64	0.61	0.67
UArizonaC	17.93 %	12.74 %	0.34	0.21	**0.92**
UNSLA	13.66 %	9.68 %	0.59	0.48	0.72
[Trotzek 2020] (a)	**12.13** %	8.77 %	0.71	0.71	0.71
[Trotzek 2020] (b)	13.52 %	**7.29** %	0.55	0.41	0.85
[Trotzek 2020] (c)	13.32 %	11.33 %	**0.73**	**0.77**	0.69
TVT	13.13 %	8.17 %	0.54	0.42	0.73
$\tau-$SS3	12.60 %	7.70 %	0.55	0.43	0.77
SS3	12.60 %	8.12 %	0.52	0.44	0.63

Dans le tableau 3.13, nous rapportons nos meilleurs modèles, les meilleurs modèles durant la tâche eRisk 2017 ainsi que les meilleurs modèles de la littérature.

Parmi les modèles des participants à eRisk 2017, nous avons retenu : FHDOA et FHDOB [Trotzek 2017] (meilleurs modèles respectivement sur $ERDE_5$ et f1-mesure), UArizonaC [Sadeque 2017] (meilleur modèle sur la mesure rappel), et enfin UNSLA [Funez 2017b] (meilleur modèle sur la mesure $ERDE_{50}$).

Parmi les résultats de la littérature, nous avons retenu le travail de Trotzek *et al.* [Trotzek 2020], qui sont aussi les auteurs de FHDOA et FHDOB, où ils ont publié plusieurs modèles, mais nous ne retenons que les trois qui ont fourni les meilleurs résultats sur les cinq mesures. Nous avons aussi retenu le travail de Funez *et al.* [Funez 2017a], appelé TVT, qui est une variante du modèle UNSLA, publié par les auteurs dans la même conférence que la tâche eRisk. Nous avons aussi considéré deux autres modèles de la littérature, $\tau-$SS3 [Burdisso 2019a] et SS3 [Burdisso 2019b], qui ont obtenu de meilleurs résultats que nos modèles sur les mesures $ERDE_5$ et $ERDE_{50}$.

En regardant les résultats, nos modèles n'ont pas obtenu les meilleurs résultats. Cependant, nos modèles ont de bonnes performances sur trois mesures qui sont la f1-mesure, la précision et le rappel où les différences avec les meilleurs résultats sont respectivement de 0.09, 0.08 et 0.05. En effet, en considérant les résultats dans le tableau 3.13 mais aussi tous les résultats des participants à la tâche, nous obtenons le deuxième meilleur résultat en considérant la mesure rappel et deuxième ex-aequo en considérant les mesures f1-mesure et précision.

Concernant les deux mesures $ERDE_5$ et $ERDE_{50}$, les différences avec les meilleurs résultats sont respectivement de 0.60 et 1.25. Les résultats sont assez décevants, mais en considérant les résultats dans le tableau 3.13 et tous les résultats des participants à la tâche, nous obtenons quand même le cinquième meilleur résultat sur ces deux mesures.

Dans le tableau 3.14, nous rapportons nos meilleurs modèles, les meilleurs modèles durant la tâche eRisk 2018 ainsi que les meilleurs modèles de la littérature sur le jeu de données eRisk 2018. Parmi les modèles des participants, nous avons retenu : UNSLA [Funez 2018] (meilleur $ERDE_5$), FHDO-BCSGB [Trotzek 2018] (meilleur $ERDE_{50}$ et f1-mesure), RKMVERIC [Paul 2018] (meilleure précision), et enfin UDCB [Cacheda 2018] (meilleur rappel).

Parmi les résultats de la littérature, nous avons retenu le travail de Burdisso *et al.* [Burdisso 2019a], appelé $\tau-$SS3. Ce travail est le meilleur dans la littérature, cependant l'article ne présente les résultats que sur les deux mesures $ERDE_5$ et $ERDE_{50}$.

Nous pouvons constater que nos modèles obtiennent les meilleurs résultats sur les mesures $ERDE_{50}$ et précision. Sur les trois autres mesures, $ERDE_5$, f1-mesure et rappel, nos modèles n'ont pas obtenu les meilleurs résultats. Sur ces trois mesures, les différences avec les meilleurs résultats sont respectivement de 0.16, 0.06 et 0.14.

TABLEAU 3.14 – Résultats de nos meilleurs modèles comparés à ceux de la littérature pour la détection au plus tôt de la dépression sur le jeu de données 2018.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModComb_moyenne (seuil simple)	8.94 %	6.31 %	0.54	0.46	0.65
ModLR (seuil complexe)	9.52 %	**6.12** %	0.51	0.38	0.80
ModRF_traits_réduits (seuil simple)	9.62 %	6.92 %	0.58	0.69	0.51
ModRF_traits_réduits (seuil complexe)	9.79 %	7.84 %	0.52	**0.71**	0.41
ModLR (seuil simple)	9.63 %	6.51 %	0.44	0.30	0.81
UNSLA	**8.78** %	7.39 %	0.38	0.48	0.32
FHDO-BCSGB	9.50 %	6.44 %	**0.64**	0.64	0.65
RKMVERIC	9.81 %	9.08 %	0.48	0.67	0.38
UDCB	15.79 %	11.95 %	0.18	0.10	**0.95**
$\tau-$SS3	9.48 %	6.17 %	-	-	-

Cependant, en considérant les résultats dans le tableau 3.14 mais aussi tous les résultats des participants à la tâche, nous obtenons le deuxième meilleur résultat en considérant la mesure rappel, troisième ex-aequo sur la mesure $ERDE_5$ et quatrième ex-aequo en considérant la mesure f1-mesure.

Bilan :

Nous avons testé nos cinq modèles sur les jeux de données eRisk 2017 et 2018. Nous observons que les modèles basés sur les caractéristiques sont les plus précis puis qu'ils offrent les meilleurs résultats sur les mesures précision et f1-mesure. Sur eRisk 2017, ces modèles offrent aussi les meilleurs résultats sur la mesure $ERDE_5$. Le modèle basé sur le plongement de phrase, quant à lui, fonctionne mieux sur les deux autres mesures $ERDE_{50}$ et rappel. Sur eRisk 2018, ce modèle produit même les meilleurs résultats sur ces mesures.

Nous avons aussi observé que la combinaison des modèles produit une amélioration sur la mesure $ERDE_{50}$ quel que soit le jeu de données. Sur le jeu de données de 2017, il y a aussi une amélioration sur le rappel alors qu'avec le jeu de données de 2018, c'est sur la mesure $ERDE_5$.

En comparant nos meilleurs modèles par rapport aux méthodes de la littérature, nous obtenons les deuxièmes meilleurs résultats en considérant les mesures rappel, précision et f1-mesure (deuxième ex-aequo sur les deux dernières mesures) sur le jeu de données eRisk 2017. Sur le jeu de données eRisk 2018, nous obtenons les meilleurs résultats en considérant les mesures $ERDE_{50}$ et précision.

3.6 Détection au plus tôt de l'anorexie

Dans cette section, nous présentons les résultats de notre approche sur la détection au plus tôt de l'anorexie ; cela nous permet d'étudier la portabilité de notre approche à d'autres tâches proches.

Nous avons utilisé les mêmes modèles que ceux présentés plus haut pour la dépression ; la différence est qu'ils ont été entraînés sur un jeu de données concernant l'anorexie. Nous avons utilisé les mêmes caractéristiques que celles présentées dans la section 3.3 et pour le plongement de phrases, il a été entraîné sur le corpus d'entraînement du jeu de données sur l'anorexie.

Nous avons donc utilisé les mêmes classifieurs que pour la détection de la dépression avec les mêmes configurations, c'est-à-dire que quand on parle de modèles qui utilisent les forêts d'arbres de décision, alors ils sont entraînés sur les vecteurs caractéristiques et les modèles qui utilisent la régression logistique sont entraînés sur les vecteurs du plongement de phrases.

Nous avons aussi effectué la réduction de caractéristiques avec la méthode "Chi-squared ranking" ; parmi les 256 caractéristiques initiales, 57 ont été sélectionnées dont 39 font partie de nos propositions. Nous avons constaté que 44 caractéristiques sont communes aux 57 caractéristiques sélectionnées sur le jeu de données sur l'anorexie, aux 124 sélectionnées sur le jeu de données de eRisk 2017 sur la dépression et aux 154 sélectionnées sur le jeu de données de eRisk 2018 sur la dépression. Plus de détails sur ces caractéristiques sélectionnées sont présentés dans le tableau 5.1 (Annexe A).

3.6.1 Résultats

Nous présentons les résultats de nos modèles sur le jeu de données eRisk 2018 sur la détection de l'anorexie. Le tableau 3.15 présente les résultats des modèles en utilisant le seuil simple alors que le tableau 3.16 présente les résultats utilisant le seuil complexe.

Comme pour la dépression, les modèles basés sur les caractéristiques offrent le meilleur résultat sur la mesure précision et le modèle basé sur le plongement de phrases offre les meilleurs résultats sur les mesures $ERDE_{50}$ et rappel.

Ce qui est particulier avec la détection de l'anorexie, c'est que nous constatons une amélioration sur la f1-mesure lorsque l'on combine les modèles utilisant les caractéristiques et le plongement de phrases ; ce qui n'a jamais eu lieu avec la dépression pour laquelle nous avons constaté une amélioration soit sur les mesures $ERDE_5$ et $ERDE_{50}$ sur le jeu de données de 2018, soit sur les mesures $ERDE_{50}$ et rappel sur le jeu de données de 2017.

TABLEAU 3.15 – Résultats avec le seuil simple pour la détection au plus tôt de l'anorexie. Les meilleurs résultats sont en gras.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_256_traits	12.81 %	9.84 %	0.54	**0.89**	0.39
ModRF_traits_réduits	12.40 %	8.60 %	0.71	**0.89**	0.59
ModLR	12.55 %	**6.48** %	0.65	0.52	**0.88**
ModComb_moyenne	**12.31** %	6.77 %	**0.74**	0.71	0.78
ModComb_max	12.59 %	6.52 %	0.65	0.51	**0.88**

TABLEAU 3.16 – Résultats avec le seuil complexe pour la détection au plus tôt de l'anorexie. Les meilleurs résultats sont en gras.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModRF_256_traits	12.85 %	11.60 %	0.50	**0.93**	0.34
ModRF_traits_réduits	12.93 %	8.91 %	0.69	0.88	0.56
ModLR	**12.53** %	**6.27** %	0.73	0.64	**0.85**
ModComb_moyenne	12.97 %	8.01 %	**0.75**	0.87	0.66
ModComb_max	12.61 %	6.35 %	0.71	0.61	**0.85**

Nous avons aussi comparé nos meilleurs modèles aux meilleurs résultats obtenus par les participants à la tâche eRisk 2018 ainsi que les meilleurs résultats de la littérature. Le tableau 3.17 présente cette comparaison.

Parmi les modèles des participants, nous avons retenu : UNSLB [Funez 2018] (meilleur $ERDE_5$), FHDO-BCSGD [Trotzek 2018] (meilleur $ERDE_{50}$ et rappel), FHDO-BCSGE [Trotzek 2018] (meilleure f1-mesure), et enfin UNSLD [Funez 2018] (meilleure précision). Parmi les résultats de la littérature, nous avons retenu le travail de Cusmuliuc *et al.* [Cusmuliuc 2019].

Nous pouvons constater que nos modèles obtiennent les meilleurs résultats sur les mesures précision et rappel (premier ex-aequo sur le rappel). Sur les trois autres mesures, $ERDE_5$, $ERDE_{50}$, et f1-mesure, nos modèles n'ont pas obtenu les meilleurs résultats. Sur ces trois mesures, les différences avec les meilleurs résultats sont respectivement de 0.91, 2.94 et 0.10. En considérant les résultats dans le tableau 3.17 mais aussi tous les résultats des participants à la tâche, nous obtenons le sixième meilleur résultat en considérant la mesure $ERDE_{50}$, le septième sur la mesure f1-mesure et le huitième en considérant la mesure $ERDE_5$.

Nous voyons aussi que notre modèle qui fournit le meilleur résultat sur la préci-

sion est le modèle entraîné sur les 256 caractéristiques alors que ces caractéristiques étaient désignées pour la dépression au départ. Ce qui signifie que des caractéristiques sont importantes pour les deux tâches. Il y a peut-être un lien entre la dépression et l'anorexie. Une analyse plus approfondie, par exemple par ablation successive de caractéristiques pourrait permettre d'aller plus loin dans l'analyse. Nous n'avons pas pu intégrer cette analyse faute de temps.

Notre méthode peut donc être utilisée pour un problème autre que la dépression à savoir la détection de l'anorexie sans effort d'ingénierie autre que le ré-apprentissage du modèle.

TABLEAU 3.17 – Résultats de nos meilleurs modèles comparés à ceux de la littérature pour la détection au plus tôt de l'anorexie.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModComb_moyenne (seuil simple)	12.31 %	6.77 %	0.74	0.71	0.78
ModLR (seuil complexe)	12.53 %	6.27 %	0.73	0.64	0.85
ModComb_moyenne (seuil complexe)	12.97 %	8.01 %	0.75	0.87	0.66
ModRF_256_traits (seuil complexe)	12.85 %	11.60 %	0.50	**0.93**	0.34
ModLR (seuil simple)	12.55 %	6.48 %	0.65	0.52	**0.88**
UNSLB	**11.40** %	7.82 %	0.61	0.75	0.51
FHDO-BCSGD	12.15 %	5.96 %	0.81	0.75	**0.88**
FHDO-BCSGE	11.98 %	6.61 %	**0.85**	0.87	0.83
UNSLD	12.93 %	9.85 %	0.79	0.91	0.71
CVMLP	12.95 %	**3.33** %	0.81	0.86	0.76
LSLDA	15.18 %	4.63 %	0.50	0.36	0.83
IDFMLP	13.08 %	5.32 %	0.68	0.76	0.61
W2VMLP	13.24 %	5.79 %	0.62	0.67	0.59

3.6.2 Intégration des termes-clés

Nous avons essayé d'intégrer les termes-clés dans un de nos modèles, plus précisément dans le modèle basé sur le plongement de phrases. Au lieu de représenter le texte tout entier en vecteur avec Doc2vec, ce sont les termes-clés [12], extraits du texte avec la méthode que nous avons présentée dans le chapitre 2 (les configurations utilisées sont

12. Nous avons extrait jusqu'à 50 termes-clés pour chaque texte. D'autres nombres de termes-clés ont été testés, mais c'est avec 50 que nous avons obtenu les meilleurs résultats.

les mêmes que pour la collection INSPEC), qui sont représentés en vecteur. Dans cet objectif, nous avons transformé chaque terme-clé extrait du texte en vecteur puis nous avons fait la moyenne de ces vecteurs pour obtenir le vecteur final. Nous avons utilisé le classifieur régression logistique. Les résultats de ce nouveau modèle sont présentés dans le tableau 3.18. Par rapport au modèle n'utilisant pas les termes-clés (ModLR), les résultats du nouveau modèle sont détériorés, sauf sur la mesure rappel où l'on observe une amélioration. Le nouveau modèle perd beaucoup en précision et en la capacité à détecter au plus tôt. Une analyse des termes-clés extraits est à mener pour expliquer ce résultat, surtout une comparaison des termes-clés issus des textes des personnes anorexiques et des personnes non-anorexiques, afin de voir la différence.

TABLEAU 3.18 – Résultats du modèle basé sur le plongement de phrases avec les termes-clés sur la détection de l'anorexie.

Modèle	$ERDE_5$	$ERDE_{50}$	F	P	R
ModLR_termes-clés (seuil complexe)	18.13 %	11.57 %	0.32	0.19	0.95
ModLR_termes-clés (seuil simple)	18.39 %	12.29 %	0.30	0.18	0.95

3.7 Conclusion

Selon l'Organisation Mondiale de la Santé (OMS), le nombre de personnes dans le monde souffrant de dépression est d'environ 300 millions de personnes. La détection de ce trouble est cruciale et constitue un défi pour la santé individuelle et publique.

Dans ce chapitre, nous avons utilisé une méthode d'apprentissage supervisée pour la détection au plus tôt de la dépression à partir des publications des utilisateurs sur les réseaux sociaux. Ce travail propose une solution pour résoudre le problème abordé dans la tâche eRisk dont l'objectif est de détecter le plus tôt possible les signes de la dépression dans les textes des utilisateurs. Pour mesurer ce facteur de temporalité, les données mises à disposition durant la tâche ont été divisées en 10 partitions où chaque partition contient 10 % des publications de chaque utilisateur. Durant la phase de test de la tâche, les partitions sont considérées de façon successive et à chaque étape il est possible de classer l'utilisateur comme dépressif ou non dépressif ou de retarder la décision afin d'utiliser des données additionnelles. Si le système proposé prend la décision de classer l'utilisateur comme dépressif ou non dépressif, la décision est définitive et ne peut plus être changée. Plus le système utilise de partitions pour prendre une décision, plus il est pénalisé. Ce travail a été validé sur les jeux de données de la

tâche eRisk, des éditions 2017 et 2018. L'édition 2018 de la tâche eRisk propose aussi de résoudre le problème de la détection au plus tôt de l'anorexie. Nous avons aussi utilisé notre méthode pour résoudre ce problème afin de voir sa portabilité sur des problèmes autres que la dépression.

Pour la résolution du problème de la tâche eRisk, nous avons utilisé des classifieurs classiques, tels que la régression logistique, utilisant en entrée : (a) des vecteurs de caractéristiques et (b) des vecteurs basés sur le plongement de phrases. Ces deux types de vecteurs sont construits à partir des publications des utilisateurs. Nous avons développé trois types de modèles : (i) utilisant seulement les vecteurs de caractéristiques, (ii) utilisant seulement les vecteurs basés sur le plongement de phrases et (iii) une combinaison des deux précédents types de modèles. Au total, nous avons proposé cinq modèles : deux de type (i), un de type (ii) et deux de type (iii).

Nous avons observé que les modèles basés sur les caractéristiques sont très performants sur la mesure précision en offrant toujours les meilleurs résultats sur cette mesure que ce soit pour la détection de la dépression (2017 et 2018) ou pour la détection de l'anorexie. Et seulement pour la dépression, ils offrent les meilleurs résultats sur la f1-mesure. Le modèle utilisant le plongement de phrases, quant à lui, est performant sur les mesures $ERDE_{50}$ et rappel. Sur les jeux de données de eRisk 2018 (pour la dépression et l'anorexie), il offre même les meilleurs résultats. Ces modèles ont donc à peu près les mêmes comportements quelle que soit la maladie mentale à détecter ou le jeu de données utilisé.

Les résultats de la combinaison des deux modèles sont différents pour chaque jeu de données. En effet pour la détection de la dépression, même si nous observons toujours une amélioration sur la mesure $ERDE_{50}$, sur le jeu de données de eRisk 2017, nous observons aussi une amélioration sur la mesure rappel (R) alors que sur le jeu de données de eRisk 2018, l'amélioration est sur la mesure $ERDE_5$. Quant à la détection de l'anorexie, la combinaison de modèles fournit une amélioration sur une autre mesure qui est la f1-mesure.

Finalement, quand nous avons comparé nos meilleurs modèles aux meilleurs résultats de la littérature (y compris les résultats des participants à la tâche eRisk) sur chaque jeu de données, nous avons obtenu les résultats suivants : meilleurs résultats avec la mesure précision sur les jeux de données eRisk 2018, que ce soit pour la détection de la dépression ou de l'anorexie. Sur le jeu de données de eRisk 2017 (dépression), nous obtenons le deuxième meilleur résultat sur cette mesure. Ces résultats sont obtenus avec nos modèles basés sur les caractéristiques. Avec le modèle basé sur le plongement de phrases, nous avons obtenu le meilleur résultat en considérant la mesure $ERDE_{50}$ sur eRisk 2018 pour la dépression et le meilleur résultat en considérant la mesure rappel

pour l'anorexie.

Nous avons aussi essayé d'intégrer l'utilisation des termes-clés dans le modèle basé sur le plongement de phrases pour la détection de l'anorexie, mais les résultats n'ont pas été concluants sauf sur la mesure rappel. Nous avons observé une amélioration sur cette mesure mais le modèle perd beaucoup en précision et en la capacité à détecter au plus tôt.

Ces travaux pourraient être complétés par l'exploration de nouvelles caractéristiques, surtout celles concernant l'anorexie dans le but d'améliorer nos modèles sur ce type de maladie. Les travaux futurs pourraient également aussi intégrer dans nos modèles d'autres données, autre que des textes, comme le nombre d'amis, nombre de j'aime, etc. présentes dans les réseaux sociaux. D'autres techniques de fusion ou de combinaison de nos modèles pourraient être étudiées, comme par exemple celle proposée par Maigrot *et al.* [Maigrot 2018] : au lieu de combiner les probabilités en sortie de nos modèles (moyenne ou maximum) comme nous le faisons, l'idée serait d'utiliser des classifieurs pour la fusion. Finalement, une étude des modèles en sortant successivement des caractéristiques pourrait être menée afin de mieux comprendre les résultats.

Les travaux présentés dans ce chapitre ont été publiés dans :

1. Faneva Ramiandrisoa et Josiane Mothe. Early Detection of Depression and Anorexia from Social Media : A Machine Learning Approach. In Proceedings of the Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020, 2020.
2. Farah Benamara, Véronique Moriceau, Josiane Mothe et Faneva Ramiandrisoa. Aide à la détection automatique des utilisateurs dépressifs dans les médias sociaux. Document Numérique, vol. 22, no. 3, pages 49–74, 2019.
3. Farah Benamara, Véronique Moriceau, Josiane Mothe, Faneva Ramiandrisoa et Zhaolong He. Automatic Detection of Depressive Users in Social Media. In COnférence en Recherche d'Informations et Applications - CORIA 2018, 15th French Information Retrieval Conference, Rennes, France, May 16-18, 2018. Proceedings, 2018.
4. Faneva Ramiandrisoa, Josiane Mothe, Farah Benamara et VéroniqueMoriceau. IRIT at e-Risk 2018. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018.
5. Idriss Abdou Malam, Mohamed Arziki, Mohammed Nezar Bellazrak, Farah Benamara, Assafa El Kaidi, Bouchra Es-Saghir, Zhaolong He, Mouad Housni, Véronique Moriceau, Josiane Mothe et Faneva Ramiandrisoa. IRIT at e-Risk. In Wor-

king Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September11-14, 2017.

Chapitre 4

Détection de textes agressifs

Sommaire

Résumé.

Les médias sociaux tels que Facebook, Twitter, etc. constituent des lieux où les gens peuvent facilement partager des informations, mais aussi d'interagir les uns avec les autres dans un espace pratiquement illimité [Samghabadi 2020a]. Ces plates-formes sont facilement accessibles et fournissent beaucoup de liberté aux utilisateurs notamment l'anonymat. Les utilisateurs peuvent donc les utiliser à mauvais escient comme pour des activités de cyber-agression ou d'agression interpersonnelle. Cela peut nuire à l'identité, au statut, à la santé mentale ou au prestige de la victime. La modération des contenus des médias sociaux est donc devenue cruciale, mais modérer les contenus manuellement est coûteux.

De nombreuses études et recherches ont été consacrées à la construction de systèmes de modération automatique afin d'éviter les discours de haine et l'agression sur les médias sociaux. Il existe aussi plusieurs forums d'évaluation qui s'intéressent à cette tâche. L'une de ces compétitions est la tâche TRAC [Kumar 2018b, Kumar 2020c] dont l'objectif est de détecter automatiquement l'agression dans les textes publiés par les utilisateurs dans les réseaux sociaux. Le travail de recherche présenté dans ce chapitre vise une solution pour résoudre le problème abordé durant la tâche TRAC. Pour cela, nous proposons d'utiliser des méthodes d'apprentissage supervisé en utilisant des classifieurs allant des classifieurs classiques (forêts d'arbres de décision, régression logistique) aux apprentissages profonds (CNN, LSTM, BERT). Pour l'évaluation de nos modèles, nous avons utilisé les collections de données des deux éditions de la tâche TRAC (2018 et 2020). Dans la première édition, deux collections de test ont été utilisées : Facebook et Twitter. Notre meilleur modèle, basé sur BERT, nous a permis d'obtenir le meilleur résultat sur la collection de test Twitter sur 30 participants et le cinquième meilleur résultat sur la collection de test Facebook. Il est important de dire que l'entraînement des modèles est uniquement réalisé sur une collection composée de contenu Facebook. La collection de test Twitter est fournie par les organisateurs pour tester le pouvoir de généralisation des modèles durant la tâche. Nous pouvons donc dire que notre modèle est généralisable. La seconde édition de TRAC est constituée de deux sous-tâches : la détection de l'agression comme durant la première édition et la détection d'agression misogyne. Avec notre meilleur modèle, qui est une combinaison de deux techniques d'apprentissage profond (CNN et LSTM), nous avons obtenu le neuvième meilleur résultat pour les deux sous-tâches, sur 16 participants à la première sous-tâche et 15 à la seconde. Donc, en général, les modèles utilisant l'apprentissage profond fournissent de meilleurs résultats que les modèles utilisant des classifieurs classiques.

4.1 Introduction

Les médias sociaux ont changé la façon dont les gens communiquent. L'un de ces aspects est la cyber-agression et l'agression interpersonnelle qui peuvent être catalysées par l'anonymat perçu sur ces plates-formes [Mishna 2018]. Des mots agressifs, des propos injurieux ou des discours de haine sont utilisés pour nuire à l'identité, au statut, à la santé mentale ou au prestige de la victime. Ce phénomène est préoccupant et il convient d'apporter des solutions permettant de le détecter, voire de le limiter.

L'agression est un sentiment de colère qui se traduit par un comportement hostile et une volonté d'attaquer [Samghabadi 2020b]. L'agression peut être exprimée de manière directe et explicite ou indirecte et sarcastique selon Kumar *et al.* [Kumar 2018c]. Le discours de haine est parfois/souvent utilisé pour attaquer une personne ou un groupe de personnes en fonction de leur couleur, sexe, race, orientation sexuelle, origine ethnique, nationalité, religion, etc. [Samghabadi 2020b].

Il est essentiel d'identifier l'agression et le discours de haine dans les réseaux sociaux pour protéger les utilisateurs contre de telles attaques, mais le faire manuellement est coûteux. Cependant, il est difficile de faire la distinction entre un contenu acceptable et un contenu agressif ou haineux en raison de la subjectivité des définitions et des différentes perceptions du même contenu par différentes personnes. Cela rend difficile la construction de systèmes automatisés. Facebook affirme même, dans son rapport d'audit sur les droits civils qui explique sa stratégie pour lutter contre les contenus abusifs et haineux, que la construction d'un système d'automatisation complet pour détecter les discours de haine n'est pas possible et que la modération des contenus est donc inévitable [Samghabadi 2020b]. Cette constatation amène de nombreux chercheurs à se concentrer sur la construction de systèmes de détection de discours de haine ou d'agression sur les réseaux sociaux. Dans la même optique, plusieurs forums d'évaluation ont aussi été organisés, dont "Abusive Language Online" (ALW) [Roberts 2019], "Trolling, Aggression and Cyberbullying" (TRAC) [Kumar 2020c] et la tâche "Semantic Evaluation" (SemEval) sur l'identification du langage offensant dans les médias sociaux (OffensEval) [Zampieri 2019].

Dans ce chapitre, nous nous intéressons à la détection automatique de l'agression dans les textes publiés par les utilisateurs dans les réseaux sociaux. Nous avons utilisé une méthode d'apprentissage supervisée pour classer si un texte contient de l'agressivité ou non. Nous avons utilisé plusieurs classifieurs allant des classifieurs classiques aux apprentissages profonds. Ce travail a été validé sur les jeux de données de la tâche TRAC des éditions 2018 et 2020. Cette tâche se focalise sur l'identification de la présence d'agressivités dans les textes (voir Section 4.4 pour plus de détails sur la tâche).

Ce travail propose donc une solution pour résoudre le problème abordé dans la tâche TRAC.

Le document est organisé comme suit : la Section 4.2 présente l'état de l'art sur les méthodes d'identification de discours de haine ou d'agression. La Section 4.3 détaille notre méthode pour la détection automatique de l'agression dans les textes. La Section 4.4 présente les jeux de données, des deux éditions de la tâche TRAC (2018 et 2020), que nous avons utilisés pour évaluer notre approche et décrit aussi la tâche TRAC. La Section 4.5 présente les résultats de notre méthode. Enfin la Section 4.6 conclut ce chapitre.

4.2 État de l'art

Dans cette section, nous discutons d'abord des travaux pertinents sur la détection de l'agression sur les médias sociaux. Ensuite, nous présentons les réseaux de neurones de type *transformers* car nous les utilisons dans l'une de nos approches.

4.2.1 Détection de l'agression

Des recherches se sont centrées sur les problèmes de détection de l'agression sur Internet. La détection de l'agression sur les médias sociaux est étroitement liée à la détection de la cyber intimidation, des discours haineux, du langage offensant et abusif.

De récents états de l'art sont présentés dans les travaux [Schmidt 2017] et [Mishra 2019]. Schmidt *et al.* [Schmidt 2017] présentent un état de l'art sur les méthodes de détection de discours haineux utilisant le traitement de langage naturel et Mishra *et al.* [Mishra 2019] rapportent un état de l'art sur les méthodes de détection automatique de langage abusif ainsi que des détails sur des jeux de données qui sont utilisés. Ces travaux rapportent que les méthodes d'apprentissage supervisé sont très largement utilisées. Les SVM et les réseaux de neurones sont les plus répandus. Ils rapportent aussi que les caractéristiques qui sont utilisées par les méthodes d'apprentissage supervisées pour la détection des discours haineux sont les caractéristiques de surface simples (exemple : sac de mots, N-grammes, etc.), la généralisation des mots (exemple : plongement de mots, etc.), et les caractéristiques basées sur les connaissances (exemple : ontologie, etc.). Mishra *et al.* [Mishra 2019] ajoutent que les récentes approches de pointe pour la détection des abus reposent sur les apprentissages profonds CNN et RNN.

Dans les différents travaux du domaine, les chercheurs ne partagent pas les jeux de données qu'ils ont collectés et utilisés [Fortuna 2018]. Par conséquent, les forums

d'évaluation, les compétitions et les tâches qui explorent ou étudient ces problèmes de détection attirent une importante attention puisqu'ils fournissent des jeux de données annotés.

Parmi ces compétitions, nous pouvons citer TRAC [Kumar 2020b] et Offenseval [Zampieri 2019]. Les nombreuses compétitions existantes couvrent diverses langues comme l'allemand [Struß 2019], l'hindi [Kumar 2018b], le bengali [Bhattacharya 2020]. Ces compétitions ne sont pas seulement différentes par rapport à la langue, mais aussi par rapport aux objectifs poursuivis. Par exemple, la détection des discours haineux envers les immigrants et les femmes [Basile 2019] ou la détection de langage abusif [Zampieri 2019].

Notre travail dans ce chapitre est validé sur les jeux de données des deux éditions de la tâche TRAC (2017 et 2018) [Kumar 2018b, Kumar 2020b] (plus de détails sur la tâche dans la section 4.4).

Durant la première édition de la tâche [Kumar 2018b], les meilleurs résultats ont été obtenus avec les méthodes d'apprentissage profond [Aroyehun 2018] sur la langue anglaise. Sur la langue hindi, les meilleurs résultats ont été obtenus avec une méthode d'apprentissage classique, la régression logistique, basée sur des caractéristiques [Samghabadi 2018]. La différence entre les modèles utilisant l'apprentissage profond et ceux utilisant les classifieurs classiques n'est cependant pas grande.

Durant la deuxième édition de la tâche TRAC [Kumar 2020b], les meilleurs résultats ont été obtenus avec BERT, un modèle de représentation de texte qui est composé de multiples couches de transformation bidirectionnelle. Nous présentons BERT dans la prochaine section.

4.2.2 BERT

BERT est un modèle de représentation de langage indépendant de la tâche à réaliser et qui se compose de plusieurs couches de transformation bidirectionnelle de Vaswani *et al.* [Vaswani 2017]. Après avoir été entraîné sur un grand corpus, il peut non seulement être utilisé pour la classification de textes mais également pour de nombreuses autres tâches, telles que la reconnaissance d'entités nommées, le résumé de texte, etc. Le pré-entraînement du modèle BERT utilise une technique appelée technique de masquage. Étant donné une phrase, 15 % des mots d'entrée sont masqués et durant l'entraînement, l'objectif consiste à prédire ces mots. Cette technique surmonte la limitation du traitement unidirectionnel et est également supérieure aux modèles de langage qui combinent le traitement de droite à gauche et de gauche à droite [Peters 2018].

L'entraînement est assez coûteux en termes de temps et d'équipement, mais un

certain nombre de modèles pré-entraînés sont disponibles librement. Il suffit alors de ré-entraîner un modèle pré-entraîné sur un petit jeu d'entraînement spécifique afin d'obtenir un nouveau modèle adapté pour la nouvelle situation. Les modèles BERT pré-entraînés par Google sont accessibles librement sur github [1]. Deux types de modèles BERT pré-entraînés sont publiés (ce sont les plus utilisés) : BERT_{Base} et BERT_{Large}. Le modèle BERT_{Base} contient 12 couches de taille 768, 12 têtes d'auto-attention (self-attention heads) et 110M paramètres, tandis que le modèle BERT_{Large} contient 24 couches de taille 1024, 16 têtes d'auto-attention et 340M paramètres.

BERT a été utilisé dans plusieurs tâches et a fourni les meilleurs résultats comme dans Offenseval [Zampieri 2019], GermEval [Struß 2019], deuxième édition de TRAC [Kumar 2020b], etc.

4.3 Méthodes

Pour la détection d'agression dans les textes, nous avons aussi utilisé une méthode d'apprentissage supervisée pour construire nos modèles. Nous avons utilisé les mêmes méthodes que nous avons utilisées dans le chapitre 3, auxquelles nous ajoutons des méthodes d'apprentissage profond. Nous avons donc au final construit huit modèles dont quatre utilisent des classifieurs classiques et quatre utilisent l'apprentissage profond et parmi ces derniers trois utilisent BERT. Ce sont les modèles avec les classifieurs classiques qui sont les mêmes que ceux proposés dans le chapitre 3 avec une petite différence pour le modèle basé sur les caractéristiques.

4.3.1 Méthode avec des classifieurs classiques

Avec les classifieurs classiques (régression logistique et les forêts d'arbres de décision), les données en entrée doivent être des réels ou des vecteurs contenant des nombres réels. Pour transformer un texte en vecteur de réels, nous avons utilisé les mêmes techniques que nous avons utilisées au chapitre 3 (section 3.3), c'est-à-dire : la représentation avec des caractéristiques (comme avec ModRF_256_traits ou ModRF_traits_reduits dans le chapitre 3) et la représentation avec le plongement de mots/-phrases (comme avec ModLR dans le chapitre 3).

Pour la représentation avec des caractéristiques, nous avons utilisé les mêmes caractéristiques qui sont présentées dans le chapitre 3 (section 3.3.1) sauf celles du groupe "Comportement de l'utilisateur" et la caractéristique "Symptômes de la dépression et

1. `https://github.com/google-research/bert`, consulté le 04 Février 2020

des noms d'antidépresseurs". Nous n'avons pas utilisé ces caractéristiques car elles sont très spécifiques aux jeux de données utilisés dans le chapitre 3. À ces caractéristiques, nous avons ajouté une caractéristique qui compte le nombre de gros mots ou d'insultes dans les textes. Nous avons réutilisé une liste de gros mots ou d'insultes utilisée dans le papier [Agrawal 2018] [2]. Au total, nous avons 246 caractéristiques.

Comme dans le modèle ModLR du Chapitre 3, pour la représentation avec le plongement de phrases, nous avons aussi utilisé Doc2Vec [Le 2014]. Nous avons entraîné deux modèles de Doc2vec sur les données d'entraînement puis nous avons concaténé les deux vecteurs représentant le texte, obtenus par les deux modèles, formant ainsi un seul vecteur de dimension 200. Les modèles de plongement de phrases Doc2vec sont entraînés sur les ensembles d'entraînement et de validation pour chaque tâche (voir section 4.4 pour plus de détails sur les jeux de données).

Avec les classifieurs classiques, nous avons créé quatre modèles : un construit avec des caractéristiques, un second utilisant le plongement de phrases et les deux derniers sont une combinaison des deux premiers (plus de détails dans la section 4.5).

4.3.2 Méthode avec l'apprentissage profond

Ce modèle combine deux techniques d'apprentissage profond : CNN et LSTM. L'idée principale est de passer la représentation d'entrée (matrice de phrase sur la figure 4.1) par un CNN et de transmettre les caractéristiques locales apprises par le CNN (vecteur concaténé sur la figure 4.1) à LSTM. En effet, CNN et LSTM sont complémentaires car chacun d'eux capture des informations à différentes échelles [Aroyehun 2018].

L'architecture de notre modèle combiné est illustrée dans la figure 4.1. Nous convertissons d'abord les phrases/textes en matrices où chaque ligne d'une matrice est une représentation vectorielle de chaque mot dans la phrase/texte que cette matrice va représenter. La dimension d'une matrice de phrases est $l \times d$, où l est la longueur du plus long du texte/phrase dans le jeu de données et d est la dimension de la représentation vectorielle des mots. La représentation vectorielle des mots est obtenue avec le modèle de plongement de mots Word2Vec [Mikolov 2013b] entraîné sur les ensembles d'entraînement et de validation pour chaque tâche.

Ensuite, des convolutions sont appliquées sur les matrices de phrases où nous avons utilisé trois types de filtres : bi-grammes (hauteur = 2), tri-grammes (hauteur = 3) et quatre-grammes (hauteur = 4). Au total, 300 filtres sont utilisés où il y a 100 filtres par type. Les résultats des convolutions s'appellent des vecteurs de caractéristiques ; les

2. `https://github.com/sweta20/Detecting-Cyberbullying-Across-SMPs`, consulté le 26 Mars 2018.

longueurs des vecteurs sont variables selon le type de filtre utilisé. Ensuite, un regroupement (1-max pooling) est effectué sur les vecteurs de caractéristiques de même taille fournissant ainsi trois vecteurs de taille 100. Il y a trois vecteurs parce qu'il y a trois types de filtres (donc trois tailles de vecteurs de caractéristiques) et de taille 100 parce qu'il y a 100 filtres par type.

Ensuite, ces trois vecteurs sont concaténés pour former un seul vecteur et un dropout est appliqué sur ce vecteur concaténé. Le vecteur concaténé est transmis à la couche LSTM. Ensuite, nous avons ajouté une couche totalement connectée (*fully connected layer*) pour réduire la dimension du vecteur résultant de la couche LSTM. Un dropout est ensuite appliqué et enfin une couche de sortie avec trois états de sortie possibles est ajoutée. Sur la couche de sortie, pour générer la classification finale, la fonction d'activation softmax est utilisée.

L'architecture de notre modèle est inspirée de l'architecture CNN proposée par *Zhang et. al.* [Zhang 2015] et qui est utilisée pour la classification de textes. Leur architecture CNN surpasse les méthodes de base telle que SVM.

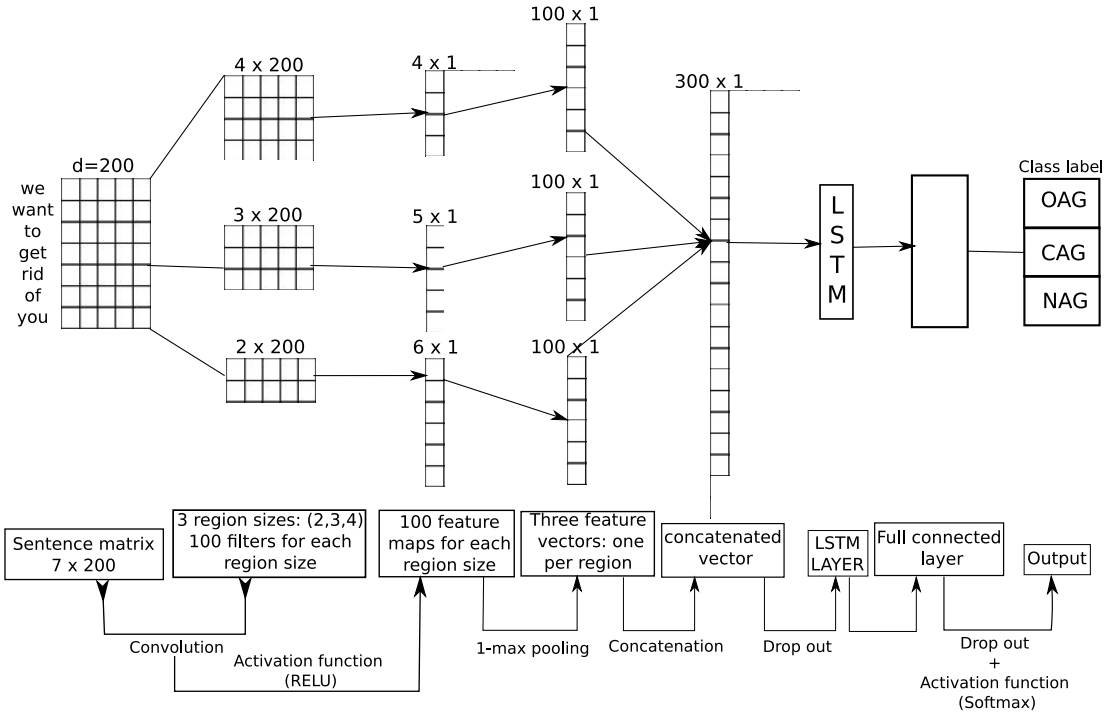


FIGURE 4.1 – Illustration de l'architecture de CNN + LSTM inspirée de [Zhang 2015].

4.3.3 Méthode avec BERT

Nous avons aussi créé des modèles avec la technique de transfert d'apprentissage en utilisant BERT. Pour cela, nous avons utilisé deux architectures de BERT : BERT original [Devlin 2019] et BERT_Pals [Stickland 2019]. BERT_Pals est une modification de BERT où est ajoutée en parallèle à BERT, une couche d'attention multi-têtes de faible dimension (Projected Attention Layers ou Couches d'Attentions Projetées (Pals)).

Dans ce qui suit, nous allons expliquer en détail le BERT original puis nous allons indiquer quelles modifications y ont été apportées pour obtenir BERT_Pals. Les détails que nous présentons dans ce qui suit sont issus du travail de Stickland *et al.* [Stickland 2019].

Le modèle BERT prend en entrée une séquence de tokens [3] et produit en sortie une représentation vectorielle de cette séquence. Un token spécial ([CLS]) est toujours insérée comme premier token de chaque séquence.

Chaque token de la séquence possède son propre vecteur caché et ces vecteurs cachés sont transformés avec la première couche BERT pour obtenir les premiers états cachés. Les premiers états cachés sont transformés successivement par des couches BERT et à la fin les états cachés finaux sont obtenus. Seul l'état caché final du token spécial [CLS] est utilisé comme représentation de la séquence en entrée pour les tâches de classification ou de régression. Le modèle BERT d'origine est simplement une pile de couches BERT successives.

Une couche BERT suit une architecture de transformations basée sur une couche d'attention multi-têtes [Vaswani 2017]. La couche multi-têtes se compose de n produits scalaires de mécanismes d'attention comme indiqué dans l'équation 4.1.

$$MH(h_j) = W^0[Attention_1(h_j), \ldots, Attention_n(h_j)] \tag{4.1}$$

où $Attention_i$ est le $i^{ème}$ mécanisme d'attention (équation 4.2) et W^0 est une matrice de taille $d \times d$.

$$Attention_i(h_j) = \sum_t softmax(\frac{W_i^q h_j \cdot W_i^k h_t}{\sqrt{d/n}}) W_i^v h_t \tag{4.2}$$

où h_j est un vecteur caché de dimension d du $j^{ème}$ token de la séquence, t parcourt chaque élément de la séquence, et W_i^q, W_i^k et W_i^v sont des matrices de taille $d/n \times d$.

La sortie de la couche d'attention multi-têtes est utilisée dans la couche d'auto-attention qui est définie dans l'équation 4.3.

3. Un token est un mot dans la phrase donnée en entrée à BERT.

$$SA(h_j) = FFN(LN(h_j + MH(h_j))) \tag{4.3}$$

où LN est une couche de normalisation [Ba 2016] et FFN est un réseau standard feed-forward (équation 4.4).

$$FFN(h_j) = W_2 f(W_1 h_j + b_1) + b_2 \tag{4.4}$$

avec f une fonction non linéaire (la fonction GELU pour BERT). W_1 et W_2 sont respectivement des matrices de taille $d_f \times d$ et $d \times d_f$, et b_1 et b_2 sont respectivement des vecteurs de dimension d_f et d.

Finalement, une couche BERT est une couche de normalisation appliquée à la sortie d'une couche d'auto-attention avec une connexion résiduelle comme indiqué dans l'équation 4.5.

$$BL(h_j) = h_j^1 = LN(h_j + SA(h_j)) \tag{4.5}$$

où h_j^1 est un état caché qui est la sortie de la première couche BERT, correspondant au vecteur caché h_j.

La formule générale d'une couche BERT est donnée par l'équation 4.6.

$$h^{l+1} = LN(h^l + SA(h^l)) \tag{4.6}$$

où l indique la couche BERT et h^{l+1} sont les états cachés qui sont des sorties de la $l^{ème}$ couche BERT. Chaque état caché est de dimension d_m sauf les h^0 car ils correspondent aux vecteurs cachés qui sont de dimension d.

Nous avons vu le modèle BERT d'origine, considérons maintenant le modèle BERT_Pals. Le modèle BERT_Pals modifie le modèle BERT d'origine en ajoutant en parallèle à chaque couche BERT une fonction spécifique. La figure 4.2 fournit une illustration des architectures des modèles BERT et BERT_Pals avec seulement deux couches pour plus de simplicité.

La nouvelle couche BERT dans le modèle BERT_Pals est alors donnée par l'équation 4.7.

$$h^{l+1} = LN(h^l + SA(h^l) + TS(h^l)) \tag{4.7}$$

où TS la fonction spécifique (équation 4.8). Si TS produit en sortie un vecteur contenant des zéros, alors nous retrouvons le modèle BERT d'origine. Si nous nous référons à la figure 4.2, $LN_l = h^l$, $SA_l = SA(h^{l-1})$ and $PAL_l = TS(h^{l-1})$.

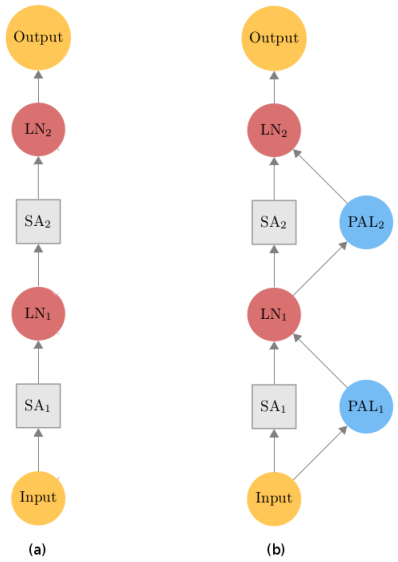


FIGURE 4.2 – Architecture de : (a) BERT et (b) BERT_Pals [Stickland 2019].

$$TS(h) = V^D g(V^E h) \tag{4.8}$$

où V^E est une matrice encodeur de taille $d_s \times d_m$, V^D est une matrice décodeur de taille $d_m \times d_s$ avec $d_s < d_m$, et g est une fonction arbitraire. La fonction g prend la forme d'une attention multi-têtes telle que les matrices V^E et V^D sont partagées entre les couches, et $d_s = 204$.

Cette combinaison de la fonction g et de d_s s'appelle Pals [Stickland 2019]. Stickland *et al.* [Stickland 2019] ont testé différentes formes de g et des valeurs de d_s, mais c'est avec Pals qu'ils ont obtenu les meilleures performances.

Pour une explication plus détaillée sur le modèle BERT_Pals, nous renvoyons les lecteurs au papier de Stickland *et al.* [Stickland 2019]. Le code source du modèle BERT_Pals est également open-source et est disponible sur github [4].

Le modèle BERT_Pals a été conçu par ses auteurs pour l'apprentissage multi-tâche mais il peut être utilisé pour l'apprentissage pour une tâche. Un apprentissage multi-tâche consiste à entraîner un seul modèle pour résoudre plusieurs tâches en l'entraînant sur plusieurs jeux de données qui sont les jeux de données des tâches à résoudre. Dans les apprentissages simples, il faut créer un modèle pour chaque tâche. Par rapport à BERT, le nombre de paramètres à estimer durant l'entraînement est moindre

4. `https://github.com/AsaCooperStickland/Bert-n-Pals`

dans BERT_Pals. BERT_Pals est donc moins coûteux pour l'apprentissage multi-tâche.

Avec BERT, nous avons créé trois modèles : un avec BERT original, un avec BERT_Pals pour un apprentissage simple et le dernier est aussi avec BERT_Pals mais pour l'apprentissage multi-tâche. Contrairement aux deux premiers modèles où nous créons un modèle pour chaque tâche en les entraînant sur chaque jeu de données d'entraînement de la tâche correspondante, pour le modèle BERT_Pals pour l'apprentissage multi-tâche, nous créons un seul modèle en l'entraînant sur tous les jeux d'entraînement.

Dans nos modèles, nous utilisons un modèle pré-entraîné sur des corpus composés de BooksCorpus (800 millions de mots) [Zhu 2015] et de Wikipédia en anglais (2 500 millions de mots). Ce modèle pré-entraîné est de type BERT_{Large}.

4.4 Tâche TRAC et jeu de données

Dans cette section, nous décrivons d'abord la tâche TRAC, ensuite, nous présentons les jeux de données des deux éditions de la tâche (2018 et 2020) puisque nous évaluons notre approche sur ces données.

4.4.1 Tâche TRAC

La tâche TRAC se concentre sur les phénomènes d'agression en ligne, de cyber-intimidation et d'autres phénomènes connexes, en particulier sur les médias sociaux. L'objectif des organisateurs est de créer une plate-forme pour des discussions académiques sur ce phénomène. Ils sont particulièrement intéressés à promouvoir des conversations dédiées à la détection automatique de l'agression dans le texte, c'est-à-dire qu'ils espèrent que TRAC sera non seulement de nature purement académique mais qu'il générera également des solutions pour lutter contre les phénomènes étudiés dans la vie réelle [Kumar 2020a]. La tâche TRAC est actuellement composée de deux éditions, en 2018 et 2020.

La première édition de TRAC comprend des données en anglais et en hindi collectées sur les plates-formes de médias sociaux Facebook et Twitter. Il s'agit d'une tâche de classification avec des textes étiquetés par trois labels au choix :

- Ouvertement Agressifs (OAG) : des textes qui expriment l'agression de manière directe et explicite.
- Secrètement Agressifs (CAG) : des textes qui expriment l'agression d'une manière indirecte et sarcastique.
- Non Agressifs (NAG) : des textes qui ne sont pas agressifs.

Durant la tâche, des publications et des commentaires collectés sur Facebook ont été fournis pour l'entraînement des systèmes des participants, tandis que, pour les tests, deux ensembles différents, un de Facebook et un de Twitter, ont été fournis.

La seconde édition de TRAC comprenait deux sous-tâches : (a) la détection de l'agression comme durant la première édition et (b) la détection d'agression misogyne. La sous-tâche (a) est en tout point identique à la première édition de la tâche alors que la sous-tâche (b) se concentre sur les agressions basées sur le genre. La sous-tâche (b) consistait en une tâche de classification binaire avec des textes étiquetés avec un des deux labels :

— Genre (GEN) : des textes qui expriment l'agression en ciblant une personne ou un groupe de personnes en fonction du sexe ou de la sexualité.
— Non Genre (NGEN) : des textes qui ne parlent pas de genre.

Cette deuxième édition comprenait des données en anglais, en hindi et en bengali, collectées sur les plates-formes de médias sociaux.

Dans nos travaux, nous n'avons considéré que les jeux de données en anglais.

4.4.2 Jeux de données TRAC 2018 et TRAC 2020

Les statistiques des jeux de données en anglais sont présentées dans les tableaux 4.1 et 4.2.

Ces jeux de données sont divisés en trois ensembles : entraînement, validation et test. Les ensembles d'entraînement et de validation sont utilisés pour entraîner des modèles.

L'ensemble d'entraînement de 2018 est composé de 11,999 textes tandis que l'ensemble de validation est composé de 3,001 textes. Pour l'ensemble de test, deux collections ont été fournies : la première est composée de 916 textes extraits de Facebook et la seconde est composée de 1,257 textes extraits de Twitter. La collection construite à partir de Twitter est ce que les organisateurs ont appelé la *collection surprise* et l'idée derrière cette collection est de tester le pouvoir de généralisation des modèles développés.

Le tableau 4.2 présente les données en anglais du jeu de données TRAC 2020. Les mêmes textes sont utilisés pour les deux sous-tâches, seules les annotations changent. Les ensembles d'entraînement et de validation sont composés de 5,000 textes collectés sur les médias sociaux alors que l'ensemble de test est composé de 1,200 textes, provenant aussi des médias sociaux.

Les données sont des textes que nous avons prétraités afin de les nettoyer. Pour chaque texte dans les jeux de données, nous avons converti tous les majuscules en mi-

TABLEAU 4.1 – Distribution des données d'entraînement, de validation et de test sur le jeu de données en anglais de la première édition de TRAC (2018).

Nombre de	Entraînement	Validation	Test	
			Facebook	Twitter
textes	11,999	3,001	916	1,257
OAG	2,708	711	144	361
CAG	4,240	1,057	142	413
NAG	5,051	1,233	630	483

TABLEAU 4.2 – Distribution des données d'entraînement, de validation et de test sur le jeu de données en anglais de la deuxième édition de TRAC (2020).

Nombre de	Entraînement	Validation	Test
textes	4,263	1,066	1,200
OAG	435	113	286
CAG	453	117	224
NAG	3,375	836	690
GEN	309	73	175
NGEN	3,954	993	1,025

nuscules, nous avons remplacé tous les liens URL par "http". Nous avons aussi converti toutes les émoticônes en texte en utilisant un outil de conversion [5]. Finalement, nous avons effacé tous les mots non encodés avec UTF-8.

4.5 Expérimentations et Résultats

Dans cette section, nous allons présenter les processus d'entraînement des modèles que nous avons utilisés pour la détection d'agression dans les textes. Nous présentons aussi les résultats des huit modèles sur les jeux de données de la tâche TRAC.

La mesure d'évaluation que nous avons considérée est la f1-mesure (F) pondérée qui est aussi la mesure utilisée durant les deux éditions de la tâche TRAC. La mesure F pondérée est égale à la moyenne de la mesure F de chaque label, pondérée par le

5. `https://github.com/carpedm20/emoji`, consulté le 04 Février 2020.

nombre de labels.

4.5.1 Processus d'entraînement

Quatre de nos modèles sont construits avec des classifieurs classiques dont un construit avec des caractéristiques, un utilisant le plongement de phrases et les deux derniers sont une combinaison des deux premiers. Pour construire les deux premiers modèles, nous avons entraîné quatre classifieurs sur les ensembles d'entraînement et les avons testés sur les ensembles de validation. Les classifieurs sont : les forêts d'arbres de décision (RF), régression logistique (RL), SVM et le bayésien naïf (NB). Le tableau 4.3 présente les résultats sur l'ensemble de validation des deux jeux de données de la tâche TRAC. Pour l'édition 2020 de la tâche, les résultats pour les deux sous-tâches sont présentés.

TABLEAU 4.3 – Résultats des classifieurs obtenus sur l'ensemble de validation des jeux de données des deux éditions de la tâche TRAC 2018 et 2020. Les meilleurs résultats sont en gras.

	Caractéristiques			Plongement de phrases		
	2018	2020		2018	2020	
		sous-tâche (a)	sous-tâche (b)		sous-tâche (a)	sous-tâche (b)
RF	**0.512**	**0.719**	**0.901**	0.439	**0.692**	**0.898**
LR	0.443	0.684	0.804	**0.498**	0.628	0.799
SVM	0.336	0.475	0.533	0.494	0.634	0.772
NB	0.449	0.433	0.457	0.444	0.672	0.787

Tout comme dans le chapitre 3, le meilleur modèle, sur les deux jeux de données (2018 et 2020), utilise le classifieur forêts d'arbres de décision (RF) quand les données sont transformées avec les caractéristiques. Par contre, quand les données sont transformées avec le plongement de phrases Doc2vec, les meilleurs résultats sont obtenus avec différents classifieurs sur les différents jeux de données : le classifieur régression logistique (RL) sur le jeu de données 2018 (comme dans le chapitre 3) et le classifieur RF sur le jeu de données 2020. Dans la suite de ce chapitre, les modèles basés sur des caractéristiques sont entraînés avec le classifieur forêts d'arbres de décision. Pour les modèles qui utilisent le plongement de phrases, ils sont entraînés avec le meilleur classifieur pour chaque jeu de données comme indiqué dans le tableau 4.3.

Comme pour les modèles présentés au chapitre 3, nous avons réduit le nombre de caractéristiques avec la méthode du "Chi-squared ranking". Sur le jeu de données 2018,

121 caractéristiques sont sélectionnées alors que sur le jeu de données 2020, ce sont 103 caractéristiques pour la sous-tâche (a) et 38 caractéristiques pour la tâche (b). Nous reportons aussi les résultats des modèles entraînés avec ces caractéristiques filtrées.

Pour le troisième et le quatrième modèles, qui sont une combinaison des deux modèles basés sur les caractéristiques et sur le plongement de phrases, nous combinons les probabilités en sortie des deux modèles. Pour cela, nous utilisons les mêmes approches que nous avons adoptées dans le chapitre 3, c'est-à-dire : la moyenne et le maximum.

Pour l'entraînement de ces classifieurs classiques, nous avons utilisé les jeux d'entraînement et de validation. Pour les autres modèles, c'est-à-dire les modèles utilisant les apprentissages profonds (CNN, LSTM et BERT), ils ont été entraînés sur les jeux d'entraînement. Les jeux de validation ont été utilisés pour l'arrêt précoce (early stopping) des entraînements des modèles. L'entraînement avec l'arrêt précoce est très utilisé et efficace dans les apprentissages profonds.

Dans la section suivante, nous présentons les résultats de nos modèles ainsi que les résultats des participants à la tâche TRAC.

4.5.2 Résultats

Dans cette section, nous présentons les résultats de notre approche sur les jeux de données de TRAC 2018 et 2020 sur la détection de l'agressivité. Nous avons, pour cela, proposé les modèles suivants dont les quatre premiers sont les mêmes que dans le chapitre 3 avec quand même une petite différence pour les deux modèles basés sur les caractéristiques.

- Mod_traits : ce modèle est construit avec le classifieur forêts d'arbres de décision (RF) en utilisant les 246 caractéristiques.
- Mod_traits_réduits : ce modèle est une variante du premier modèle et est aussi construit avec le classifieur RF, mais cette fois-ci en utilisant les caractéristiques filtrées.
- Mod_Doc2vec : ce modèle est construit en utilisant le plongement de phrases. Ce modèle utilise le classifieur régression logistique (RL) sur les jeux de données de l'édition 2018 de la tâche TRAC et le classifieur RF sur les jeux de données de l'édition 2020.
- ModComb_moyenne : ce modèle est la combinaison des deux modèles : Mod_traits_réduits et Mod_Doc2vec. Pour cela, la probabilité en sortie de ce modèle est la moyenne des probabilités en sortie des deux modèles qui le composent.
- ModComb_max : ce modèle est aussi une combinaison des deux modèles : Mo-

dRF_traits_réduits et Mod_Doc2vec. Ce modèle prend comme probabilité en sortie, le maximum des probabilités en sortie des deux modèles qui le composent.
— Mod_CNN_LSTM : ce modèle combine deux techniques d'apprentissage profond : CNN et LSTM.
— Mod_BERT : ce modèle est construit en utilisant BERT, la version originale.
— Mod_BERT_Pals : ce modèle est construit en utilisant une version modifiée BERT_Pals.
— Mod_BERT_Pals_multi : ce modèle est aussi construit avec BERT_Pals avec l'apprentissage multi-tâche. C'est-à-dire que ce modèle est entraîné sur les tous jeux de données des deux éditions de TRAC.

Le tableau 4.4 présente les résultats de nos modèles sur les deux données de test (Facebook et Twitter) de l'édition 2018 de TRAC. Nous y voyons aussi les résultats des deux participants qui ont été les meilleurs sur chaque donnée de test. Notons que ces deux participants utilisent l'apprentissage profond pour créer leurs modèles.

Nous pouvons constater que parmi nos modèles, le meilleur résultat est obtenu avec le modèle Mod_BERT que ce soit sur la collection de test Facebook ou la collection de test Twitter. La différence entre le meilleur modèle et le moins bon est de 0.087 sur la collection de test Facebook et 0.391 sur la collection de test Twitter. Tous les modèles fonctionnent mieux sur la collection de test Facebook que sur l'autre collection (Twitter). Cela est peut-être dû au fait que l'entraînement des modèles est réalisé sur une collection semblable à la collection de test Facebook.

En regardant les modèles utilisant les classifieurs classiques, qui sont des modèles créés avec les mêmes techniques que pour la détection de la dépression et de l'anorexie dans le chapitre 3, nous voyons que les modèles utilisant les caractéristiques sont meilleurs que le modèle utilisant le plongement de phrase sur la collection de test Facebook et vice-versa sur la collection de test Twitter. De plus, le filtrage de caractéristiques mais aussi la combinaison de modèles (avec l'approche maximum) que nous avons effectués améliorent seulement les résultats sur la collection de test Facebook. Nous pouvons constater que ces modèles utilisant les classifieurs classiques fonctionnent quand même pour la détection de l'agression puisque la différence avec le meilleur résultat est de 0.017 (avec le modèle ModComb_max) sur la collection de test Facebook et 0.036 (avec le modèle Mod_Doc2vec) sur la collection de test Twitter.

Les modèles d'apprentissage profond basés sur BERT fournissent les meilleurs résultats. Nous constatons cependant que l'apprentissage multi-tâche fonctionne moins bien que l'apprentissage avec le BERT original. Une analyse plus profonde doit être menée pour expliquer cela. Le modèle BERT_Pals obtient les moins bons résultats parmi les modèles basés sur BERT ; d'ailleurs la valeur sur la collection Twitter de ce mo-

dèle est étrangement basse. Notre hypothèse est que le modèle BERT_Pals fonctionne mieux avec l'apprentissage multi-tâche que dans le cas d'une simple tâche ; cela se vérifie avec les résultats du modèle Mod_BERT_Pals_multi.

En comparant les résultats du meilleur modèle (Mod_BERT) aux résultats des participants à l'édition 2018 de TRAC, nous obtenons le meilleur résultat sur la collection de test Twitter et le cinquième meilleur résultat sur la collection de test Facebook sur 30 participants.

TABLEAU 4.4 – Résultats de nos meilleurs modèles comparés avec ceux de la littérature sur les données de test Facebook et Twitter de la tâche TRAC 2018.

Modèle	F (pondéré)	
	Facebook	Twitter
Mod_traits	0.568	0.429
Mod_traits_réduits	0.582	0.425
Mod_Doc2vec	0.539	0.468
ModComb_moyenne	0.575	0.453
ModComb_max	0.590	0.466
Mod_CNN_LSTM	0.520	0.520
Mod_BERT	0.607	**0.604**
Mod_BERT_Pals	0.561	0.213
Mod_BERT_Pals_multi	0.580	0.544
Saroyehun [Aroyehun 2018]	**0.642**	0.592
vista.ue [Raiyani 2018]	0.581	0.601

Le tableau 4.5 présente les résultats de nos modèles sur les données de test des deux sous-tâches de l'édition 2020 de TRAC. Nous y voyons aussi les résultats des deux participants qui ont été les meilleurs sur chaque sous-tâche. Ces deux participants utilisent l'apprentissage profond pour créer leurs modèles. Plus précisément, leurs modèles sont basés sur BERT.

C'est toujours un modèle utilisant l'apprentissage profond qui obtient les meilleurs résultats. Si sur le jeu de données 2018, c'était le modèle basé sur BERT, ici sur le jeu de données 2020, c'est le modèle Mod_CNN_LSTM.

Parmi les modèles utilisant les classifieurs classiques, tout comme sur le jeu de données 2018 (sur la collection Facebook), les modèles utilisant les caractéristiques fournissent de meilleurs résultats que le modèle utilisant le plongement de phrase. Par contre, le filtrage de caractéristiques mais aussi la combinaison de modèles n'amé-

liorent plus les résultats. Sur la sous-tâche (a), qui est semblable à la tâche 2018, les résultats sont inférieurs à ceux de 2018. C'est sûrement dû aux jeux de données qui sont différents. En effet, le nombre de textes dans le jeu de données de 2018 est plus grand que celui du jeu de données de 2020 (environ trois fois plus).

En regardant les modèles utilisant l'apprentissage profond basé sur BERT, l'apprentissage multi-tâche est meilleur que l'apprentissage simple avec BERT (original ou avec Pals), ce qui n'est pas le cas sur les données de l'édition 2018. Notre hypothèse pour expliquer cela se trouve dans les données. En effet, durant l'édition 2020 de la tâche TRAC, dans les données des deux sous-tâches, les textes sont les mêmes, seules les annotations (classes ou labels) changent. Ainsi, durant l'apprentissage multi-tâche du modèle, les textes composant les données de TRAC 2020 sont passés deux fois par le modèle alors les textes composant les données de TRAC 2018 ne passent qu'une seule fois. Une analyse plus profonde doit être menée pour vérifier cette hypothèse. Les résultats des modèles basés sur BERT sur la sous-tâche (a) sont aussi inférieurs (sauf pour la multi-tâche) à ceux sur le jeu de données de 2018. Cela est aussi peut-être dû aux jeux de données.

Nous observons quand même une exception avec notre meilleur modèle sur ce jeu de données 2020, qui est la combinaison de deux techniques d'apprentissage profond CNN et LSTM. En effet, le modèle Mod_CNN_LSTM fournit de meilleurs résultats sur la sous-tâche (a) que sur le jeu de données 2018. Une étude plus approfondie est à mener pour expliquer ceci, surtout sur les jeux de données.

En comparant les résultats du meilleur modèle, c'est-à-dire Mod_CNN_LSTM, par rapport aux résultats des participants à l'édition 2020 de la tâche TRAC, nous obtenons le neuvième meilleur résultat pour les deux sous-tâches. Il y avait 16 participants pour la sous-tâche (a) et 15 pour la sous-tâche (b). Nous avons constaté toutefois que la performance de notre modèle est plus proche du meilleur sur la sous-tâche (b) que sur la sous-tâche (a). La différence par rapport au meilleur modèle est de 0.1 pour la sous-tâche (a) et 0.025 pour la sous-tâche (b). En fait, ce sont tous nos modèles qui fonctionnent mieux sur la sous-tâche (b) que sur la sous-tâche (a).

4.5.3 Analyse des Matrices de confusion

Les figures 4.3 et 4.4 montrent les matrices de confusion normalisées de notre meilleur modèle sur les collections de test des éditions 2018 et 2020 de TRAC, BERT original pour 2018 et la combinaison de CNN et LSTM pour 2020.

En 2018, sur la collection Facebook, l'erreur de classification la plus fréquente est la prédiction de CAG (Secrètement Agressifs) au lieu de NAG (Non Agressifs) : 35 %

TABLEAU 4.5 – Résultats de nos meilleurs modèles comparés avec ceux de la littérature sur les données de test des deux sous-tâches de la tâche TRAC 2020.

Modèle	F (pondéré)	
	Sous-tâche (a)	Sous-tâche (b)
Mod_traits	0.504	0.798
Mod_traits_réduits	0.509	0.786
Mod_Doc2vec	0.421	0.786
ModComb_moyenne	0.431	0.789
ModComb_max	0.436	0.789
Mod_CNN_LSTM	0.702	0.846
Mod_BERT	0.420	0.831
Mod_BERT_Pals	0.420	0.787
Mod_BERT_Pals_multi	0.691	0.846
Julian [Risch 2020]	**0.802**	0.851
Ms8qQxMbnjJMgYcw [Gordeev 2020]	0.756	**0.871**

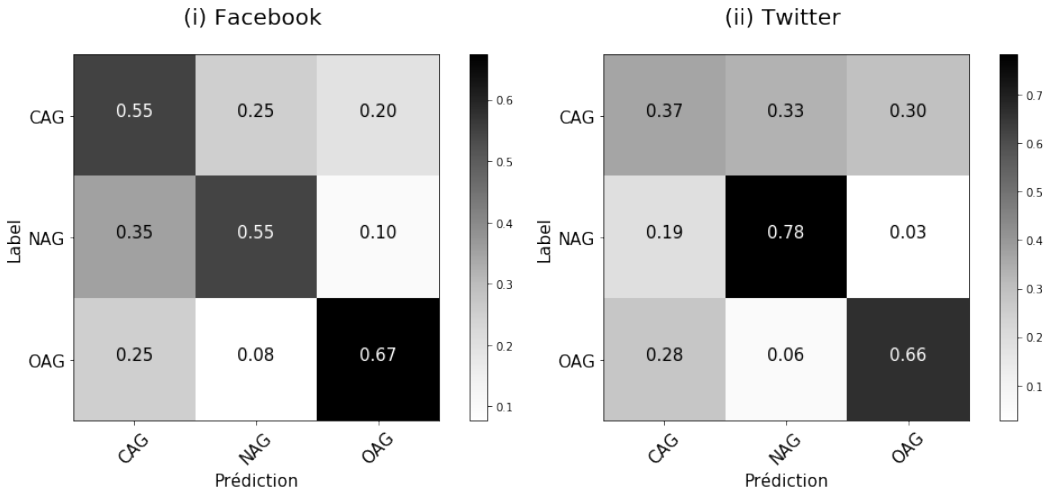


FIGURE 4.3 – Matrice de confusion de notre meilleur modèle (Mod_BERT) sur les collections de test de TRAC 2018 : (i) Facebook et (ii) Twitter.

des NAG sont prédits comme des CAG. Sur la collection Twitter, à l'inverse l'erreur la plus fréquente est la prédiction de NAG au lieu de CAG : 33 % des CAG sont prédits comme des NAG. Ainsi, sur les deux collections, notre modèle confond les deux labels

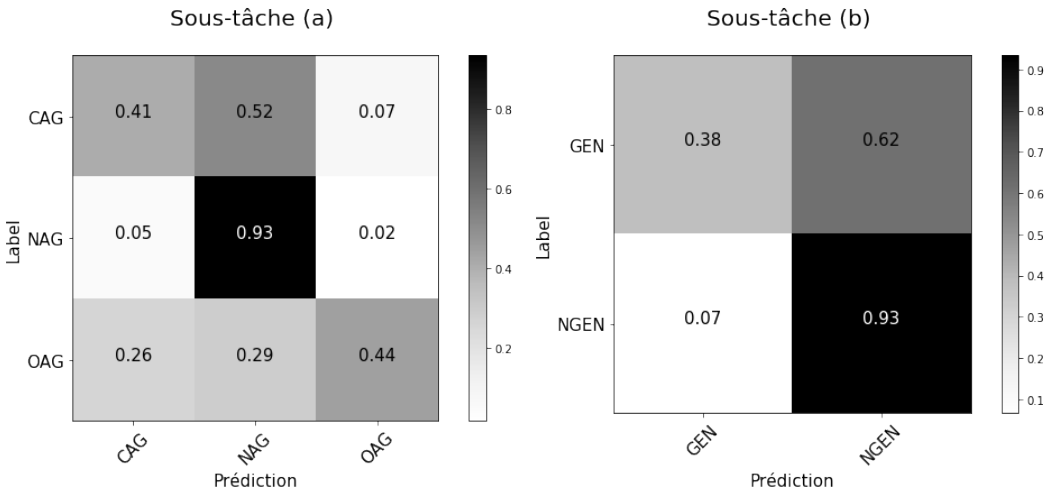


FIGURE 4.4 – Matrice de confusion de notre meilleur modèle (Mod_CNN_LSTM) pour les sous-tâches (a) détection de l'agression et (b) détection d'agression misogyne, de l'édition 2020 de TRAC.

CAG et NAG et c'est aussi le cas entre les labels CAG et OAG (Ouvertement Agressifs). Ce résultat est peut-être dû au fait que parmi les textes que nous avons lus, les textes étiquetés comme NAG sont plus similaires aux textes étiquetés comme CAG qu'à ceux étiquetés comme OAG. De plus, les textes étiquetés comme OAG sont plus similaires à ceux étiquetés comme CAG qu'à ceux étiquetés comme NAG. En effet, il est plus facile de distinguer une publication qui exprime l'agression de manière directe et explicite que celle qui l'exprime de façon indirecte et sarcastique. Il est aussi intéressant de voir que notre modèle est meilleur pour prédire OAG sur la collection Facebook alors que ce label est moins fréquent que les deux autres labels dans les collections d'entraînement et de validation de TRAC 2018 (1.5 fois moins fréquent que CAG et 1.8 fois moins fréquent que NAG).

Pour la sous-tâche (a), l'erreur de classification la plus fréquente est la prédiction de NAG au lieu de CAG, comme sur la collection Twitter du jeu de données 2018, mais ici 52 % des CAG sont prédits comme des NAG. Notre modèle est meilleur pour prédire NAG ce qui est tout à fait normal puisqu'il est environ huit fois plus nombreux que les deux autres labels (chacun). C'est peut-être cela qui a causé la difficulté du modèle à prédire CAG et OAG. En effet, ici on constate que le modèle a plus de mal à distinguer OAG de NAG contrairement à ce qui se passe avec ce même modèle sur les jeux de données de TRAC 2018 (voir la figure dans Annexe C). Une analyse plus profonde est à mener pour expliquer ce phénomène.

Pour la sous-tâche (b), notre modèle a beaucoup de difficulté à prédire le label GEN où 68 % des GEN sont prédits comme des NGEN. Cela est peut-être dû à la nature du jeu de données qui n'est pas équilibré. En effet, il y a environ trente fois plus de labels NGEN que de labels GEN dans le jeu de données.

4.6 Conclusion

Avec l'existence de plusieurs plates-formes de médias sociaux qui sont facilement accessibles et qui fournissent beaucoup de liberté aux utilisateurs, notamment l'anonymat, la modération des contenus est cruciale afin de détecter, voire contrôler des contenus non souhaités.

Dans ce chapitre, nous avons utilisé une méthode d'apprentissage supervisée pour la détection de l'agressivité dans les textes publiés par les utilisateurs sur les plates-formes de médias sociaux. Ce travail propose une solution pour résoudre le problème abordé dans la tâche TRAC. Pour cela, nous avons utilisé plusieurs classifieurs allant des classifieurs classiques (forêts d'arbres de décision et régression logistique) aux apprentissages profonds (CNN, LSTM et BERT). Plus précisément, nous avons construit huit modèles qui utilisent soit des caractéristiques, soit le plongement de phrase, soit les deux. Par ailleurs, certains de nos modèles s'appuient sur des apprentissages classiques (régression logistique, forêts d'arbres de décision) alors que les autres sont des variantes de BERT. Pour créer les modèles utilisant les classifieurs classiques, nous avons utilisé les mêmes techniques que nous avons utilisées quand nous avons créé nos modèles pour la détection de la dépression et de l'anorexie dans le chapitre 3.

Un travail de comparaison de ces modèles a été mené sur les jeux de données de la tâche de détection de l'agressivité TRAC, des éditions 2018 et 2020. Sur l'édition 2018, en considérant les résultats de tous les participants à la tâche, nous obtenons le meilleur résultat sur la collection de test Twitter et le cinquième meilleur résultat sur la collection de test Facebook. C'est le modèle basé sur la version originale de BERT qui obtient ces résultats. Il est important de noter que l'entraînement de nos modèles est réalisé sur une collection d'entraînement composée de textes extraits de Facebook. Notre modèle semble avoir un bon pouvoir de généralisation puisqu'il fonctionne bien sur la collection Twitter.

Sur l'édition 2020, nous obtenons le neuvième meilleur résultat pour les deux sous-tâches. Il y avait 16 participants pour la sous-tâche (a) et 15 pour la sous-tâche (b). C'est le modèle qui combine deux techniques d'apprentissage profond qui obtient ces résultats. Nous avons constaté toutefois que la performance de ce modèle est plus proche du meilleur sur la sous-tâche (b) que sur la sous-tâche (a). En général, tous nos modèles

fonctionnent mieux sur la sous-tâche (b) que sur la sous-tâche (a).

Les travaux présentés dans ce chapitre pourraient être étendus de plusieurs façons : nous pourrions mener une analyse afin d'essayer d'expliquer pourquoi tous nos modèles fonctionnent mieux sur la sous-tâche (b). Nous pourrions également rendre nos modèles plus robustes aux jeux de données non équilibrés comme c'est le cas des jeux de données de la tâche TRAC. Enfin, nous pourrions tester nos modèles sur d'autre données issues d'autres plates-formes de médias sociaux afin de vérifier leur pouvoir de généralisation avec une approche multi-tâche.

Ces travaux ont été publiés dans :

1. Faneva Ramiandrisoa et Josiane Mothe. IRIT at TRAC 2020. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 49–54, 2020.
2. Faneva Ramiandrisoa et Josiane Mothe. Aggression Identification in Social Media : a Transfer Learning Based Approach. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 26–31, 2020.
3. Faneva Ramiandrisoa. Aggression Identification in Posts – two machine learning approaches. In Workshop on Machine Learning for Trend and Weak Signal Detection in Social Networks and Social Media., Toulouse, France, 2020. CEUR-WS.org.
4. Faneva Ramiandrisoa et Josiane Mothe. IRIT at TRAC 2018. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018, pages 19–27, 2018.

Chapitre 5

Conclusion et perspectives

Les travaux que nous avons présentés dans cette thèse portent sur des tâches ayant une finalité applicative : détection de la dépression et de l'anorexie d'une part et détection de l'agressivité d'autre part ; cela à partir de messages postés par des utilisateurs sur les réseaux sociaux. Nous avons également proposé une amélioration d'une méthode non supervisée d'extraction de termes-clés. Ces trois groupes de travaux ont chacun fait l'objet de publications, mais ont été initiés dans des moments différents du travail de thèse compte tenu de leur relative indépendance. Lorsque possible, nous avons toutefois montré les liens ou les ponts possibles mais non réalisés faute de temps.

La première contribution de cette thèse est l'amélioration de la méthode non supervisée d'extraction automatique à base de graphes proposée par Boudin *et al.* [Boudin 2013a]. En effet, les méthodes non supervisées à base de graphes présentent généralement les faiblesses suivantes :

(1) la construction du graphe de mots est basée sur la co-occurrence qui ne capture pas très bien la relation sémantique entre deux mots.
(2) l'attribution de scores aux termes-clés candidats est en fonction de leur longueur ce qui favorise souvent les candidats les plus longs ou les plus courts.
(3) les scores des termes-clés candidats sont calculés en fonction des scores des mots qui les composent ; cela favorise la présence de chevauchements des termes-clés. Ainsi certains candidats sont des sous-chaînes d'autres candidats.
(4) la construction du graphe de mots basée sur la co-occurrence entraîne aussi la perte d'information alors que cette information peut être très importante dans le choix des termes-clés (par exemple la fréquence ou la position des termes-clés candidats dans le document).

Afin de résoudre ces problèmes, nous avons proposé de combiner différentes solutions proposées dans la littérature. Pour le problème (1), nous avons proposé d'utiliser le plongement de mots lors de la construction du graphe de mots ; celui-ci capture en effet les relations entre les mots. Pour le problème (2), nous avons adopté une modification de la moyenne harmonique [Yeom 2019] pour calculer les scores des termes-clés candidats. Enfin pour résoudre les problèmes (3) et (4), nous avons recalculé les scores des termes-clés candidats en utilisant la méthode modifiée C-value proposée par Yeom

et al..

Pour l'évaluation, nous avons utilisé onze collections de données dont cinq contiennent des documents longs, quatre contiennent des documents courts et deux contiennent des documents de type article de presse. Nous avons montré que les améliorations que nous avons apportées améliorent significativement la méthode de Boudin *et al.* [Boudin 2013a] en considérant la f1-mesure sur tous les corpus de documents longs, seulement sur un corpus de documents courts, et aucune amélioration sur le corpus documents de type article de presse. Par rapport à douze autres méthodes non supervisées de l'état de l'art, nous avons obtenu des résultats équivalents aux meilleures méthodes sur deux des cinq corpus de documents longs et sur un corpus de documents courts, le deuxième résultat sur deux autres corpus de documents longs, deux corpus de documents courts et un corpus de documents de type article de presse, et enfin le troisième résultat sur le dernier corpus de documents longs et corpus de documents courts.

Notre deuxième contribution dans cette thèse est de proposer une solution pour la détection au plus tôt de la dépression à partir des publications des utilisateurs sur les réseaux sociaux. L'objectif dans cette partie de recherche est le même que celui abordé durant la tâche eRisk qui est de détecter si un utilisateur est dépressif, en utilisant le moins de publications possible de cet utilisateur. Pour mesurer le facteur de temporalité, les publications mises à disposition durant la tâche ont été divisées en 10 partitions où chaque partition contient 10 % des publications de chaque utilisateur. Durant la phase de test de la tâche, les partitions sont considérées de façon successive et à chaque étape il est possible de classer l'utilisateur comme dépressif ou non dépressif ou de retarder la décision afin d'utiliser des données additionnelles. Si le système proposé prend la décision de classer l'utilisateur comme dépressif ou non dépressif, la décision est définitive et ne peut plus être changée. Plus le système utilise de partitions pour prendre une décision, plus il est pénalisé. Pour atteindre cet objectif, nous avons proposé des modèles utilisant des classifieurs, s'appuyant sur la régression logistique ou les forêts d'arbres de décision, basés sur (a) des caractéristiques et (b) le plongement de phrases. Ce travail a été validé sur les jeux de données de la tâche eRisk, des éditions 2017 et 2018. L'édition 2018 de la tâche eRisk propose aussi de résoudre le problème de la détection au plus tôt de l'anorexie. Ainsi, nous avons aussi utilisé nos modèles pour détecter ce problème afin de voir leur portabilité sur des problèmes autres que la dépression.

Nous avons observé que les modèles basés sur les caractéristiques sont très performants lorsque la mesure de précision est considérée en offrant toujours les meilleurs résultats, que ce soit pour la détection de la dépression ou pour la détection de l'anorexie.

Seulement pour la dépression, ils offrent les meilleurs résultats lorsque l'on considère la f1-mesure. Le modèle utilisant le plongement de phrases, quant à lui, est performant sur les mesures $ERDE_{50}$ et rappel. Sur les jeux de données de eRisk 2018 (pour la dépression et l'anorexie), il offre même les meilleurs résultats. Ces modèles ont donc à peu près les mêmes comportements quelle que soit la maladie mentale à détecter (après un entraînement adapté à la collection) ou le jeu de données utilisé. En comparant nos meilleurs modèles aux meilleurs résultats de la littérature, y compris ceux des participants à la tâche eRisk, sur chaque jeu de données, nous avons obtenu les résultats suivants : les modèles basés sur les caractéristiques obtiennent les meilleurs résultats avec la mesure précision sur les jeux de données eRisk 2018, que ce soit pour la détection de la dépression ou de l'anorexie. Sur le jeu de données de eRisk 2017 (dépression), ils obtiennent le deuxième meilleur résultat (aussi sur la précision). Avec le modèle basé sur le plongement de phrases, nous avons obtenu le meilleur résultat en considérant la mesure $ERDE_{50}$ sur eRisk 2018 pour la dépression et le meilleur résultat en considérant la mesure rappel pour l'anorexie.

La troisième et dernière contribution de cette thèse concerne la détection de l'agressivité dans les textes postés par des utilisateurs sur les réseaux sociaux. L'objectif dans cette recherche est celui proposé durant la tâche TRAC, à savoir de classer si une publication d'un utilisateur contient un texte agressif ou non. Pour répondre à cet objectif, nous avons réutilisé les mêmes modèles que ceux utilisés pour la détection de la dépression ou de l'anorexie lors de la deuxième contribution de cette thèse. À cela, nous avons ajouté d'autres modèles basés sur l'apprentissage profond (CNN, LSTM et BERT).

Pour l'évaluation de ces modèles, les collections de données des deux éditions de la tâche internationale TRAC (2018 et 2020) ont été utilisées. Nous avons observé que nos modèles, utilisant l'apprentissage profond, fournissent de meilleurs résultats que nos modèles utilisant des classifieurs classiques. Nos résultats dans cette partie de la thèse sont moyens comparés à l'état de l'art du domaine. Nous avons toutefois obtenu le meilleur résultat sur une des collections de données.

Perspectives

Comme travaux futurs pour l'extraction automatique de termes-clés, nous prévoyons d'explorer d'autres méthodes de plongement de mots/termes/phrases ou de représentation de texte telles que BERT [Devlin 2019]. Ensuite, nous aimerions explorer d'autres méthodes d'extraction de termes-clés candidats qui garantissent l'extraction de bons candidats. Une autre perspective à long terme serait d'étudier comment intégrer les groupes de mots dans les modèles comme dans BERT par exemple. Nos premières tentatives n'ont pas donné de résultats à la hauteur de nos espérances, mais nous pensons qu'un apprentissage adéquat sur les groupes de mots pourrait améliorer

la modélisation.

Pour la détection au plus tôt de la dépression ou de l'anorexie, nos travaux pourraient être étendus par l'exploration de nouvelles caractéristiques, en particulier celles concernant l'anorexie dans le but d'améliorer nos modèles sur ce type de maladie mentale. Également, la prochaine étape est d'intégrer dans nos modèles d'autres données, autre que des textes, comme le nombre d'amis, nombre de j'aime, etc. Aussi, d'autres techniques de fusion ou combinaison de nos modèles pourraient être étudiées, comme par exemple celle utilisée dans Maigrot *et al.* [Maigrot 2018] : plutôt que de combiner les probabilités en sortie de nos modèles comme nous le faisons, la fusion des prédictions pourrait être réalisée par d'autres classifieurs. Finalement, une étude des modèles en sortant successivement des caractéristiques pourrait être menée afin de mieux comprendre les résultats.

Pour la détection de l'agressivité dans les textes, nos travaux pourraient être complétés de plusieurs façons : nous pourrions mener une analyse afin d'essayer d'expliquer pourquoi tous nos modèles fonctionnent mieux sur la sous-tâche (b) de l'édition 2020 de la tâche TRAC que sur les autres tâches. Nous pourrions également rendre nos modèles plus robustes aux jeux de données non équilibrés comme c'est le cas des jeux de données de TRAC. Finalement, nous pourrions tester nos modèles sur d'autres données issues d'autres plates-formes de médias sociaux afin de vérifier leur pouvoir de généralisation avec une approche multi-tâche.

Bibliographie

[Agrawal 2018] Sweta Agrawal et Amit Awekar. Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms. In Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings, pages 141–153, 2018.

[Alzaidy 2019] Rabah Alzaidy, Cornelia Caragea et C. Lee Giles. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. In The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pages 2551–2557, 2019.

[Arora 2017] Sanjeev Arora, Yingyu Liang et Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[Aroyehun 2018] Segun Taofeek Aroyehun et Alexander F. Gelbukh. Aggression Detection in Social Media : Using Deep Neural Networks, Data Augmentation, and Pseudo Labeling. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018, pages 90–97, 2018.

[Ba 2016] Lei Jimmy Ba, Jamie Ryan Kiros et Geoffrey E. Hinton. Layer Normalization. CoRR, vol. abs/1607.06450, 2016.

[Baroni 2014] Marco Baroni, Georgiana Dinu et Germán Kruszewski. Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1 : Long Papers, pages 238–247, 2014.

[Basaldella 2018] Marco Basaldella, Elisa Antolli, Giuseppe Serra et Carlo Tasso. Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction. In Digital Libraries and Multimedia Archives - 14th Italian Research Conference on Digital Libraries, IRCDL 2018, Udine, Italy, January 25-26, 2018, Proceedings, pages 180–187, 2018.

[Basile 2019] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso et Manuela Sanguinetti. SemEval-2019 Task 5 : Multilingual Detection of Hate Speech Against

Immigrants and Women in Twitter. In Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019, pages 54–63, 2019.

[Bavelas 1950] Alex Bavelas. Communication patterns in task-oriented groups. Journal of the acoustical society of America, 1950.

[Bengio 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent et Christian Janvin. A Neural Probabilistic Language Model. Journal of Machine Learning Research, vol. 3, pages 1137–1155, 2003.

[Bennani-Smires 2018] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl et Martin Jaggi. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018, pages 221–229, 2018.

[Bhattacharya 2020] Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri et Atul Kr. Ojha. Developing a Multilingual Annotated Corpus of Misogyny and Aggression, 2020.

[Blei 2003] David M. Blei, Andrew Y. Ng et Michael I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, vol. 3, pages 993–1022, 2003.

[Bonacich 1987] Phillip Bonacich. Power and centrality : A family of measures. American journal of sociology, pages 1170–1182, 1987.

[Boudin 2013a] Florian Boudin. A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. In Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013, pages 834–838, 2013.

[Boudin 2013b] Florian Boudin. TALN Archives : a digital archive of French research articles in Natural Language Processing (TALN Archives : une archive numérique francophone des articles de recherche en Traitement Automatique de la Langue) [in French]. In Traitement Automatique des Langues Naturelles, TALN 2013, Les Sables d'Olonne, France, 17-21 Juin 2013, articles courts, pages 507–514, 2013.

[Boudin 2016] Florian Boudin. pke : an open source python-based keyphrase extraction toolkit. In COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, December 11-16, 2016, Osaka, Japan, pages 69–73, 2016.

[Boudin 2018] Florian Boudin. Unsupervised Keyphrase Extraction with Multipartite Graphs. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 667–672, 2018.

[Boudin 2020] Florian Boudin, Ygor Gallina et Akiko Aizawa. Keyphrase Generation for Scientific Document Retrieval. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 1118–1126, 2020.

[Bougouin 2013a] Adrien Bougouin. State of the Art of Automatic Keyphrase Extraction Methods (État de l'art des méthodes d'extraction automatique de termes-clés) [in French]. In Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues, RÉCITAL 2013, Les Sables d'Olonne, France, 17-21 Juin 2013, pages 96–109, 2013.

[Bougouin 2013b] Adrien Bougouin, Florian Boudin et Béatrice Daille. TopicRank : Graph-Based Topic Ranking for Keyphrase Extraction. In Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013, pages 543–551, 2013.

[Bougouin 2014] Adrien Bougouin, Florian Boudin et Béatrice Daille. The impact of domains for Keyphrase extraction (Influence des domaines de spécialité dans l'extraction de termes-clés) [in French]. In Traitement Automatique des Langues Naturelles, TALN 2014, Marseille, France, 1-4 Juillet 2014, articles longs, pages 13–24, 2014.

[Bougouin 2016] Adrien Bougouin, Florian Boudin et Béatrice Daille. Keyphrase Annotation with Graph Co-Ranking. In COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference : Technical Papers, December 11-16, 2016, Osaka, Japan, pages 2945–2955, 2016.

[Brin 1998] Sergey Brin et Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks, vol. 30, no. 1-7, pages 107–117, 1998.

[Burdisso 2019a] Sergio G. Burdisso, Marcelo Errecalde et Manuel Montes-y-Gómez. t-SS3 : a text classifier with dynamic n-grams for early risk detection over text streams. CoRR, vol. abs/1911.06147, 2019.

[Burdisso 2019b] Sergio G. Burdisso, Marcelo Errecalde et Manuel Montes-y-Gómez. A text classification framework for simple and effective early depression

detection over social media streams. Expert Syst. Appl., vol. 133, pages 182–197, 2019.

[Cacheda 2018] Fidel Cacheda, Diego Fernández Iglesias, Francisco Javier Nóvoa et Victor Carneiro. Analysis and Experiments on Early Detection of Depression. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018, 2018.

[Campos 2018] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes et Adam Jatowt. YAKE ! Collection-Independent Automatic Keyword Extractor. In Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings, pages 806–810, 2018.

[Caragea 2014] Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea et Sujatha Das Gollapalli. Citation-Enhanced Keyphrase Extraction from Research Papers : A Supervised Approach. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1435–1446, 2014.

[Carbonell 1998] Jaime G. Carbonell et Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In SIGIR '98 : Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia, pages 335–336, 1998.

[Chen 2018] Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan et Zhoujun Li. Keyphrase Generation with Correlation Constraints. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 4057–4066, 2018.

[Choudhury 2013] Munmun De Choudhury, Michael Gamon, Scott Counts et Eric Horvitz. Predicting Depression via Social Media. In Proceedings of the Seventh International Conference on Weblogs and Social Media, 2013.

[Choudhury 2014] Munmun De Choudhury, Scott Counts, Eric Horvitz et Aaron Hoff. Characterizing and predicting postpartum depression from shared facebook data. In Computer Supported Cooperative Work, CSCW '14, Baltimore, MD, USA, February 15-19, 2014, pages 626–638, 2014.

[Collobert 2008] Ronan Collobert et Jason Weston. A unified architecture for natural language processing : deep neural networks with multitask learning. In Ma-

chine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008, pages 160–167, 2008.

[Colombo 2016] Gualtiero B. Colombo, Pete Burnap, Andrei Hodorog et Jonathan Scourfield. Analysing the connectivity and communication of suicidal users on twitter. Computer Communications, vol. 73, pages 291–300, 2016.

[Cusmuliuc 2019] Ciprian-Gabriel Cusmuliuc, Lucia-Georgiana Coca et Adrian Iftene. Early Detection of Signs of Anorexia in Social Media. -, 2019.

[Dalloux 2020] Clément Dalloux, Vincent Claveau, Marc Cuggia, Guillaume Bouzillé et Natalia Grabar. Supervised Learning for the ICD-10 Coding of French Clinical Narratives. In Digital Personalized Health and Medicine - Proceedings of MIE 2020, Medical Informatics Europe, Geneva, Switzerland, April 28 - May 1, 2020, pages 427–431, 2020.

[Danesh 2015] Soheil Danesh, Tamara Sumner et James H. Martin. SGRank : Combining Statistical and Graphical Methods to Improve the State of the Art in Unsupervised Keyphrase Extraction. In Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, *SEM 2015, June 4-5, 2015, Denver, Colorado, USA, pages 117–126, 2015.

[Denker 1990] John S. Denker et Yann LeCun. Transforming Neural-Net Output Levels to Probability Distributions. In Advances in Neural Information Processing Systems 3, [NIPS Conference, Denver, Colorado, USA, November 26-29, 1990], pages 853–859, 1990.

[Devlin 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee et Kristina Toutanova. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

[Eichler 2010] Kathrin Eichler et Günter Neumann. DFKI KeyWE : Ranking Keyphrases Extracted from Scientific Articles. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pages 150–153, 2010.

[El-Beltagy 2010] Samhaa R. El-Beltagy et Ahmed A. Rafea. KP-Miner : Participation in SemEval-2. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pages 190–193, 2010.

[Ercan 2007] Gonenc Ercan et Ilyas Cicekli. Using lexical chains for keyword extraction. Inf. Process. Manage., vol. 43, no. 6, pages 1705–1714, 2007.

[Florescu 2017] Corina Florescu et Cornelia Caragea. PositionRank : An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1 : Long Papers, pages 1105–1115, 2017.

[Florescu 2018] Corina Florescu et Wei Jin. Learning Feature Representations for Keyphrase Extraction. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 8077–8078, 2018.

[Fortuna 2018] Paula Fortuna et Sérgio Nunes. A Survey on Automatic Detection of Hate Speech in Text. ACM Comput. Surv., vol. 51, no. 4, pages 85 :1–85 :30, 2018.

[France 2000] Daniel J. France, Richard G. Shiavi, Stephen E. Silverman, Marilyn K. Silverman et D. Mitchell Wilkes. Acoustical properties of speech as indicators of depression and suicidal risk. IEEE Trans. Biomed. Engineering, vol. 47, no. 7, pages 829–837, 2000.

[Frank 1999] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin et Craig G. Nevill-Manning. Domain-Specific Keyphrase Extraction. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages, pages 668–673, 1999.

[Freeman 1977] Linton C. Freeman. A set of measures of centrality based on betweenness. Sociometry, pages 35–41, 1977.

[Fuhr 2017] Norbert Fuhr. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. SIGIR Forum, vol. 51, no. 3, pages 32–41, 2017.

[Funez 2017a] Darío Gustavo Funez, Marcelo Luis Errecalde, Maria Paula Villegas, Maria José Garciarena Ucelay et Leticia Cecilia Cagnina. Temporal Variation of Terms as Concept Space for Early Risk Prediction. In Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017, 2017.

[Funez 2017b] Darío Gustavo Funez, Maria Paula Villegas, Maria José Garciarena Ucelay, Leticia Cecilia Cagnina et Marcelo Luis Errecalde. LIDIC - UNSL's

Participation at eRisk 2017 : Pilot Task on Early Detection of Depression. In Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017, 2017.

[Funez 2018] Darío Gustavo Funez, Maria José Garciarena Ucelay, Maria Paula Villegas, Sergio Burdisso, Leticia C. Cagnina, Manuel Montes-y-Gómez et Marcelo Errecalde. UNSL's participation at eRisk 2018 Lab. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018., 2018.

[Gollapalli 2014] Sujatha Das Gollapalli et Cornelia Caragea. Extracting Keyphrases from Research Papers Using Citation Networks. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada, pages 1629–1635, 2014.

[Gollapalli 2017] Sujatha Das Gollapalli, Xiaoli Li et Peng Yang. Incorporating Expert Knowledge into Keyphrase Extraction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pages 3180–3187, 2017.

[Gordeev 2020] Denis Gordeev et Olga Lykova. BERT of all trades, master of some. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 93–98, 2020.

[Hasan 2010] Kazi Saidul Hasan et Vincent Ng. Conundrums in Unsupervised Keyphrase Extraction : Making Sense of the State-of-the-Art. In COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China, pages 365–373, 2010.

[Hasan 2014] Kazi Saidul Hasan et Vincent Ng. Automatic Keyphrase Extraction : A Survey of the State of the Art. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1 : Long Papers, pages 1262–1273, 2014.

[Hulth 2003] Anette Hulth. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, Sapporo, Japan, July 11-12, 2003, 2003.

[Jiang 2009] Xin Jiang, Yunhua Hu et Hang Li. A ranking approach to keyphrase extraction. In Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009, pages 756–757, 2009.

[Joulin 2017] Armand Joulin, Edouard Grave, Piotr Bojanowski et Tomas Mikolov. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2 : Short Papers, pages 427–431, 2017.

[Kim 2010] Su Nam Kim, Olena Medelyan, Min-Yen Kan et Timothy Baldwin. SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pages 21–26, 2010.

[Kleinberg 1999] Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. J. ACM, vol. 46, no. 5, pages 604–632, 1999.

[Krapivin 2009] Mikalai Krapivin, Aliaksandr Autaeu et Maurizio Marchese. Large dataset for keyphrases extraction. Rapport technique, University of Trento, 2009.

[Kumar 2018a] Kulkarni Akshay Bhavani Kumar. Early detection of depression. Master's thesis, University of Houston, 2018.

[Kumar 2018b] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi et Marcos Zampieri. Benchmarking Aggression Identification in Social Media. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018, pages 1–11, 2018.

[Kumar 2018c] Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia et Tushar Maheshwari. Aggression-annotated Corpus of Hindi-English Code-mixed Data. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018, 2018.

[Kumar 2020a] Ritesh Kumar, Atul Kr. Ojha, Bornini Lahiri, Marcos Zampieri, Shervin Malmasi, Vanessa Murdock et Daniel Kadar, editeurs. Proceedings of the second workshop on trolling, aggression and cyberbullying, trac@lrec 2020, marseille, france, may 2020. European Language Resources Association (ELRA), 2020.

[Kumar 2020b] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi et Marcos Zampieri. Evaluating Aggression and Misogyny Identification in Social Media. In Ritesh Kumar, Atul Kr. Ojha, Bornini Lahiri, Marcos Zampieri, Shervin Malmasi, Vanessa Murdock et Daniel Kadar, editeurs, Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020), Paris, France, may 2020. European Language Resources Association (ELRA).

[Kumar 2020c] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi et Marcos Zampieri. Evaluating Aggression Identification in Social Media. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 1–5, 2020.

[Le 2014] Quoc V. Le et Tomas Mikolov. Distributed Representations of Sentences and Documents. In Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, pages 1188–1196, 2014.

[Liu 2009] Zhiyuan Liu, Peng Li, Yabin Zheng et Maosong Sun. Clustering to Find Exemplar Terms for Keyphrase Extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 257–266, 2009.

[Liu 2010] Zhiyuan Liu, Wenyi Huang, Yabin Zheng et Maosong Sun. Automatic Keyphrase Extraction via Topic Decomposition. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 366–376, 2010.

[Lopez 2010] Patrice Lopez et Laurent Romary. HUMB : Automatic Key Term Extraction from Scientific Articles in GROBID. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pages 248–251, 2010.

[Losada 2016] David E. Losada et Fabio Crestani. A Test Collection for Research on Depression and Language Use. In Experimental IR Meets Multilinguality, Multimodality, and Interaction - 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings, pages 28–39, 2016.

[Losada 2017] David E. Losada, Fabio Crestani et Javier Parapar. eRISK 2017 : CLEF Lab on Early Risk Prediction on the Internet : Experimental Foundations. In Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings, pages 346–360, 2017.

[Losada 2018] David E. Losada, Fabio Crestani et Javier Parapar. Overview of eRisk : Early Risk Prediction on the Internet. In Experimental IR Meets Multilinguality, Multimodality, and Interaction - 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings, pages 343–361, 2018.

[Low 2011] Lu-Shih Alex Low, Namunu C. Maddage, Margaret Lech, Lisa Sheeber et Nicholas B. Allen. Detection of Clinical Depression in Adolescents' Speech During Family Interactions. IEEE Trans. Biomed. Engineering, vol. 58, no. 3, pages 574–586, 2011.

[Mahata 2018] Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah et Roger Zimmermann. Key2Vec : Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 634–639, 2018.

[Maigrot 2018] Cédric Maigrot, Ewa Kijak et Vincent Claveau. Fusion par apprentissage pour la détection de fausses informations dans les réseaux sociaux. Document Numérique, vol. 21, no. 3, pages 55–80, 2018.

[Marriott 2014] Tamsin C. Marriott et Tom Buchanan. The true self online : Personality correlates of preference for self-expression online, and observer ratings of personality online and offline. Computers in Human Behavior, vol. 32, pages 171–177, 2014.

[Marujo 2012] Luís Marujo, Anatole Gershman, Jaime G. Carbonell, Robert E. Frederking et João Paulo Neto. Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization. In Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012, pages 399–403, 2012.

[Matsuo 2004] Yutaka Matsuo et Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, vol. 13, no. 1, pages 157–169, 2004.

[Medelyan 2009] Olena Medelyan, Eibe Frank et Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1318–1327, 2009.

[Meng 2017] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky et Yu Chi. Deep Keyphrase Generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1 : Long Papers, pages 582–592, 2017.

[Mihalcea 2004] Rada Mihalcea et Paul Tarau. TextRank : Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain, pages 404–411, 2004.

[Mikolov 2013a] Tomas Mikolov, Kai Chen, Greg Corrado et Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013.

[Mikolov 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado et Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Advances in Neural Information Processing Systems 26 : 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 3111–3119, 2013.

[Mishna 2018] Faye Mishna, Cheryl Regehr, Ashley Lacombe-Duncan, Joanne Daciuk, Gwendolyn Fearing et Melissa Van Wert. Social media, cyber-aggression and student mental health on a university campus. Journal of mental health, vol. 27, no. 3, pages 222–229, 2018.

[Mishra 2019] Pushkar Mishra, Helen Yannakoudakis et Ekaterina Shutova. Tackling Online Abuse : A Survey of Automated Abuse Detection Methods. CoRR, vol. abs/1908.06024, 2019.

[Mohammad 2013] Saif Mohammad et Peter D. Turney. Crowdsourcing a Word-Emotion Association Lexicon. Computational Intelligence, vol. 29, no. 3, pages 436–465, 2013.

[Mowery 2016] Danielle L. Mowery, Albert Park, Craig Bryan et Mike Conway. Towards Automatically Classifying Depressive Symptoms from Twitter Data for Population Health. In Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, PEOPLES@COLING 2016, Osaka, Japan, December 12, 2016, pages 182–191, 2016.

[Nguyen 2007] Thuy Dung Nguyen et Min-Yen Kan. Keyphrase Extraction in Scientific Publications. In Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, 10th International Conference on Asian Digital Libraries, ICADL 2007, Hanoi, Vietnam, December 10-13, 2007, Proceedings, pages 317–326, 2007.

[Nguyen 2010] Thuy Dung Nguyen et Minh-Thang Luong. WINGNUS : Keyphrase Extraction Utilizing Document Logical Structure. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pages 166–169, 2010.

[Organization 2017] World Health Organizationet al. Depression and other common mental disorders : global health estimates. 2017. Geneva : WHO, 2017.

[Ozdas 2004] Asli Ozdas, Richard G. Shiavi, Stephen E. Silverman, Marilyn K. Silverman et D. Mitchell Wilkes. Investigation of vocal jitter and glottal flow spectrum as possible cues for depression and near-term suicidal risk. IEEE Trans. Biomed. Engineering, vol. 51, no. 9, pages 1530–1540, 2004.

[Paukkeri 2010] Mari-Sanna Paukkeri et Timo Honkela. Likey : Unsupervised Language-Independent Keyphrase Extraction. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pages 162–165, 2010.

[Paul 2018] Sayanta Paul, Sree Kalyani Jandhyala et Tanmay Basu. Early Detection of Signs of Anorexia and Depression Over Social Media using Effective Machine Learning Frameworks. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018, 2018.

[Pennebaker 2003] James W. Pennebaker, Matthias R. Mehl et Kate G. Niederhoffer. Psychological aspects of natural language use : Our words, our selves. Annual review of psychology, vol. 54, no. 1, pages 547–577, 2003.

[Pennington 2014] Jeffrey Pennington, Richard Socher et Christopher D. Manning. Glove : Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1532–1543, 2014.

[Peters 2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee et Luke Zettlemoyer. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 2227–2237, 2018.

[Raiyani 2018] Kashyap Raiyani, Teresa Gonçalves, Paulo Quaresma et Vítor Beires Nogueira. Fully Connected Neural Network with Advance Preprocessor to Identify Aggression over Facebook and Twitter. In Proceedings of the First

Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018, pages 28–41, 2018.

[Resnik 2015] Philip Resnik, William Armstrong, Leonardo Max Batista Claudino, Thang Nguyen, Viet-An Nguyen et Jordan L. Boyd-Graber. Beyond LDA : Exploring Supervised Topic Modeling for Depression-Related Language in Twitter. In Proceedings of CLPsych@NAACL-HLT, 2015.

[Risch 2020] Julian Risch et Ralf Krestel. Bagging BERT Models for Robust Aggression Identification. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 55–61, 2020.

[Roberts 2019] Sarah T. Roberts, Joel Tetreault, Vinodkumar Prabhakaran et Zeerak Waseem, editeurs. Proceedings of the third workshop on abusive language online, Florence, Italy, aug 2019. Association for Computational Linguistics.

[Rose 2010] S Rutherfoord Rose, Dave Engel, Nick Cramer et Wendy Cowley. Automatic keyword extraction from individual documents. In Text Mining : Applications and Theory, 2010.

[Rude 2004] Stephanie Rude, Eva-Maria Gortner et James Pennebaker. Language use of depressed and depression-vulnerable college students. Cognition & Emotion, vol. 18, no. 8, pages 1121–1133, 2004.

[Rush 2015] Alexander M. Rush, Sumit Chopra et Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 379–389, 2015.

[Sadeque 2017] Farig Sadeque, Dongfang Xu et Steven Bethard. UArizona at the CLEF eRisk 2017 Pilot Task : Linear and Recurrent Models for Early Depression Detection. In Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017, 2017.

[Sagen 2010] Ulrike Sagen, Arnstein Finset, Torbjørn Moum, Tore Mørland, Tom Gunnar Vik, Tibor Nagy et Toril Dammen. Early detection of patients at risk for anxiety, depression and apathy after stroke. General hospital psychiatry, vol. 32, no. 1, pages 80–85, 2010.

[Salton 1975] Gerard Salton, A. Wong et Chung-Shu Yang. A Vector Space Model for Automatic Indexing. Commun. ACM, vol. 18, no. 11, pages 613–620, 1975.

[Samghabadi 2018] Niloofar Safi Samghabadi, Deepthi Mave, Sudipta Kar et Thamar Solorio. RiTUAL-UH at TRAC 2018 Shared Task : Aggression Identification. In

Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018, pages 12–18, 2018.

[Samghabadi 2020a] Niloofar Safi Samghabadi, Adrián Pastor López-Monroy et Thamar Solorio. Detecting Early Signs of Cyberbullying in Social Media. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 144–149, 2020.

[Samghabadi 2020b] Niloofar Safi Samghabadi, Parth Patwa, PYKL Srinivas, Prerana Mukherjee, Amitava Das et Thamar Solorio. Aggression and Misogyny Detection using BERT : A Multi-Task Approach. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020, pages 126–131, 2020.

[Sarkar 2010] Kamal Sarkar, Mita Nasipuri et Suranjan Ghose. A New Approach to Keyphrase Extraction Using Neural Networks. CoRR, vol. abs/1004.3274, 2010.

[Schmidt 2017] Anna Schmidt et Michael Wiegand. A Survey on Hate Speech Detection using Natural Language Processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, SocialNLP@EACL 2017, Valencia, Spain, April 3, 2017, pages 1–10, 2017.

[Schutz 2008] Alexander Thorsten Schutzet al. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. Masters of Applied Science M. App. Sc, 2008.

[Schwartz 2014] H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Gregory J. Park, Maarten Sap, David Stillwell, Michal Kosinski et Lyle H. Ungar. Towards assessing changes in degree of depression through Facebook. In Proceedings of the Workshop on Computational Linguistics and Clinical Psychology : From Linguistic Signal to Clinical Reality, pages 118–125, 2014.

[Schwenk 2007] Holger Schwenk. Continuous space language models. Computer Speech & Language, vol. 21, no. 3, pages 492–518, 2007.

[Sterckx 2015a] Lucas Sterckx, Thomas Demeester, Johannes Deleu et Chris Develder. Topical Word Importance for Fast Keyphrase Extraction. In Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pages 121–122, 2015.

[Sterckx 2015b] Lucas Sterckx, Thomas Demeester, Johannes Deleu et Chris Develder. When Topic Models Disagree : Keyphrase Extraction with Multiple Topic Models. In Proceedings of the 24th International Conference on World Wide

Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pages 123–124, 2015.

[Stickland 2019] Asa Cooper Stickland et Iain Murray. BERT and PALs : Projected Attention Layers for Efficient Adaptation in Multi-Task Learning. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 5986–5995, 2019.

[Struß 2019] Julia Maria Struß, Melanie Siegel, Josef Ruppenhofer, Michael Wiegand et Manfred Klenner. Overview of GermEval Task 2, 2019 Shared Task on the Identification of Offensive Language. In Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019, 2019.

[Sun 2020] Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang et Chaoran Zhang. SIFRank : A New Baseline for Unsupervised Keyphrase Extraction Based on Pre-Trained Language Model. IEEE Access, vol. 8, pages 10896–10906, 2020.

[Teneva 2017] Nedelina Teneva et Weiwei Cheng. Salience Rank : Efficient Keyphrase Extraction with Topic Modeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2 : Short Papers, pages 530–535, 2017.

[Trotzek 2017] Marcel Trotzek, Sven Koitka et Christoph M. Friedrich. Linguistic Metadata Augmented Classifiers at the CLEF 2017 Task for Early Detection of Depression. In Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017., 2017.

[Trotzek 2018] Marcel Trotzek, Sven Koitka et Christoph M. Friedrich. Word Embeddings and Linguistic Metadata at the CLEF 2018 Tasks for Early Detection of Depression and Anorexia. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018., 2018.

[Trotzek 2020] Marcel Trotzek, Sven Koitka et Christoph M. Friedrich. Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences. IEEE Trans. Knowl. Data Eng., vol. 32, no. 3, pages 588–601, 2020.

[Turian 2010] Joseph P. Turian, Lev-Arie Ratinov et Yoshua Bengio. Word Representations : A Simple and General Method for Semi-Supervised Learning. In ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, pages 384–394, 2010.

[Turney 2000] Peter D. Turney. Learning Algorithms for Keyphrase Extraction. Inf. Retr., vol. 2, no. 4, pages 303–336, 2000.

[Turney 2001] Peter D. Turney. Mining the Web for Synonyms : PMI-IR versus LSA on TOEFL. In Machine Learning : EMCL 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings, pages 491–502, 2001.

[Turney 2003] Peter D. Turney. Coherent Keyphrase Extraction via Web Mining. In IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003, pages 434–442, 2003.

[Vaswani 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser et Illia Polosukhin. Attention is All you Need. In Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pages 5998–6008, 2017.

[Wan 2008] Xiaojun Wan et Jianguo Xiao. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008, pages 855–860, 2008.

[Wang 2013] Xinyu Wang, Chunhong Zhang, Yang Ji, Li Sun, Leijia Wu et Zhana Bao. A Depression Detection Model Based on Sentiment Analysis in Micro-blog Social Network. In Trends and Applications in Knowledge Discovery and Data Mining - PAKDD 2013 International Workshops : DMApps, DANTH, QIMIE, BDM, CDA, CloudSD, Gold Coast, QLD, Australia, April 14-17, 2013, Revised Selected Papers, pages 201–213, 2013.

[Wang 2015] Rui Wang, Wei Liu et Chris McDonald. Using Word Embeddings to Enhance Keyword Identification for Scientific Publications. In Databases Theory and Applications - 26th Australasian Database Conference, ADC 2015, Melbourne, VIC, Australia, June 4-7, 2015. Proceedings, pages 257–268, 2015.

[Wang 2017] Liang Wang et Sujian Li. PKU_ICL at SemEval-2017 Task 10 : Keyphrase Extraction with Model Ensemble and External Knowledge. In Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017, pages 934–937, 2017.

[Wang 2018] Yanan Wang, Qi Liu, Chuan Qin, Tong Xu, Yijun Wang, Enhong Chen et Hui Xiong. Exploiting Topic-Based Adversarial Neural Network for Cross-Domain Keyphrase Extraction. In IEEE International Conference on

Data Mining, ICDM 2018, Singapore, November 17-20, 2018, pages 597–606, 2018.

[Witten 1999] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin et Craig G. Nevill-Manning. KEA : Practical Automatic Keyphrase Extraction. In Proceedings of the Fourth ACM conference on Digital Libraries, August 11-14, 1999, Berkeley, CA, USA, pages 254–255, 1999.

[Won 2019] Miguel Won, Bruno Martins et Filipa Raimundo. Automatic extraction of relevant keyphrases for the study of issue competition. In Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing, Berkeley, La Rochelle, France, April 7-13, 2019, 2019.

[Xue 2014] Yuanyuan Xue, Qi Li, Li Jin, Ling Feng, David A. Clifton et Gari D. Clifford. Detecting Adolescent Psychological Pressures from Micro-Blog. In Health Information Science - Third International Conference, HIS 2014, Shenzhen, China, April 22-23, 2014. Proceedings, pages 83–94, 2014.

[Yang 2018] Min Yang, Yuzhi Liang, Wei Zhao, Wei Xu, Jia Zhu et Qiang Qu. Task-oriented keyphrase extraction from social media. Multimedia Tools Appl., vol. 77, no. 3, pages 3171–3187, 2018.

[Ye 2018] Hai Ye et Lu Wang. Semi-Supervised Learning for Neural Keyphrase Generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 4142–4153, 2018.

[Yeom 2019] Hongseon Yeom, Youngjoong Ko et Jungyun Seo. Unsupervised-learning-based keyphrase extraction from a single document by the effective combination of the graph-based model and the modified C-value method. Computer Speech & Language, vol. 58, pages 304–318, 2019.

[Zampieri 2019] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra et Ritesh Kumar. SemEval-2019 Task 6 : Identifying and Categorizing Offensive Language in Social Media (OffensEval). In Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019, pages 75–86, 2019.

[Zhang 2006] Kuo Zhang, Hui Xu, Jie Tang et Juan-Zi Li. Keyword Extraction Using Support Vector Machine. In Advances in Web-Age Information Management, 7th International Conference,WAIM 2006, Hong Kong, China, June 17-19, 2006, Proceedings, pages 85–96, 2006.

[Zhang 2015] Ye Zhang et Byron C. Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. CoRR, vol. abs/1510.03820, 2015.

[Zhang 2016] Qi Zhang, Yang Wang, Yeyun Gong et Xuanjing Huang. Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 836–845, 2016.

[Zhu 2015] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba et Sanja Fidler. Aligning Books and Movies : Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 19–27, 2015.

Annexes

Annexe A

TABLEAU 5.1 – Liste des caractéristiques retenues sur chaque jeu de données : dépression 2017 et 2018, et anorexie.

Dépression		Anorexie
2017	2018	
Caractéristiques propres à chaque jeu de données		
empath_disgust	Fréquences des 25 5-grammes	Saison de publications (saison 1)
	Pronoms 1ère personne "mine"	empath_attractive
	Fréquence du passé	empath_competing
	Verbes conjugués au passé	empath_cooking
	empath_achievement	empath_eating
	empath_appearance	empath_fun
	empath_banking	empath_internet
	empath_beach	empath_medieval
	empath_celebration	empath_messaging
	empath_communication	empath_musical
	empath_competing	empath_power
	empath_domestic_work	empath_restaurant
	empath_dominant_heirarchical	empath_weakness
	empath_envy	
	empath_farming	
	empath_fight	
	empath_internet	
	empath_kill	
	empath_meeting	
	empath_monster	

TABLEAU 5.1 – suite du tableau de la page précédente

Dépression		Anorexie
2017	2018	
Caractéristiques propres à chaque jeu de données (suite)		
	empath_ocean empath_plant empath_power empath_sailing empath_science empath_stealing empath_traveling empath_valuable empath_war empath_weakness empath_wealthy	
Caractéristiques communes aux collections sur la dépression		
Fréquences des 25 tri-grammes Fréquence des adverbes Nombre moyen de publications Lisibilité d'un texte (DCR) Fréquence des émoticônes Fréquence des pronoms à la première personne "me" Lisibilité d'un texte (FOG) Lisibilité d'un texte (LWR) Nombre minimum de publications Fréquence de la Négation Nombre moyen de mots par publications Saison de publications (saison 2 : Mars, Avril et Mai) Sac de mots "even" Sac de mots "going"		

TABLEAU 5.1 – suite du tableau de la page précédente

Dépression		Anorexie
2017	2018	
Caractéristiques communes aux collections sur la dépression (suite)		
Sac de mots "know" Sac de mots "life" Sac de mots "like" Sac de mots "people" Sac de mots "someone" Sac de mots "things" Sac de mots "though" Sac de mots "time" Sac de mots "work" Fréquence des verbes Auxiliaires "être" conjugués au passé empath_air_travel empath_anger empath_car empath_cheerfulness empath_childish empath_children empath_computer empath_crime empath_driving empath_economics empath_emotional empath_exasperation empath_family empath_friends empath_government		

TABLEAU 5.1 – suite du tableau de la page précédente

Dépression		Anorexie
2017	2018	
Caractéristiques communes aux collections sur la dépression (suite)		
empath_hate empath_healing empath_hearing empath_help empath_home empath_horror empath_injury empath_irritability empath_journalism empath_joy empath_leader empath_listen empath_lust empath_military empath_money empath_negative_emotion empath_night empath_optimism empath_party empath_payment empath_politeness empath_politics empath_positive_emotion empath_prison empath_programming empath_real_estate		

TABLEAU 5.1 – suite du tableau de la page précédente

Dépression		Anorexie
2017	2018	
Caractéristiques communes aux collections sur la dépression (suite)		
empath_royalty empath_sexual empath_ship empath_speaking empath_technology empath_timidity empath_tool empath_trust empath_vehicle empath_violence empath_wedding empath_youth empath_zest		
Caractéristiques communes à toutes les collections		
Fréquence des mots en majuscules Noms chimiques et marques des antidépresseurs Lisibilité d'un texte (FRE) Fréquence du mot "depress" Fréquence des pronoms à la première personne "I" Fréquence des pronoms à la première personne "Im" Fréquence des pronoms à la première personne "my" Fréquence des pronoms à la première personne "Myself" Somme des fréquences des pronoms Fréquence du pronom "I" dans un contexte subjectif Fréquence des noms Nombre moyen de mots par commentaires		

TABLEAU 5.1 – suite du tableau de la page précédente

Dépression		Anorexie
2017	2018	
Caractéristiques communes à toutes les collections (suite)		
Fréquence des mots liés au sommeil		
Sac de mots "im"		
Sac de mots "day"		
Sac de mots "feel"		
Sac de mots "help"		
Sac de mots "ive"		
Sac de mots "much"		
Sac de mots "really"		
Symptômes de la dépression et des noms d'antidépresseurs		
empath_affection		
empath_body		
empath_cold		
empath_contentment		
empath_fear		
empath_feminine		
empath_health		
empath_love		
empath_medical_emergency		
empath_morning		
empath_neglect		
empath_negotiate		
empath_nervousness		
empath_pain		
empath_sadness		
empath_shame		
empath_sleep		

TABLEAU 5.1 – suite du tableau de la page précédente

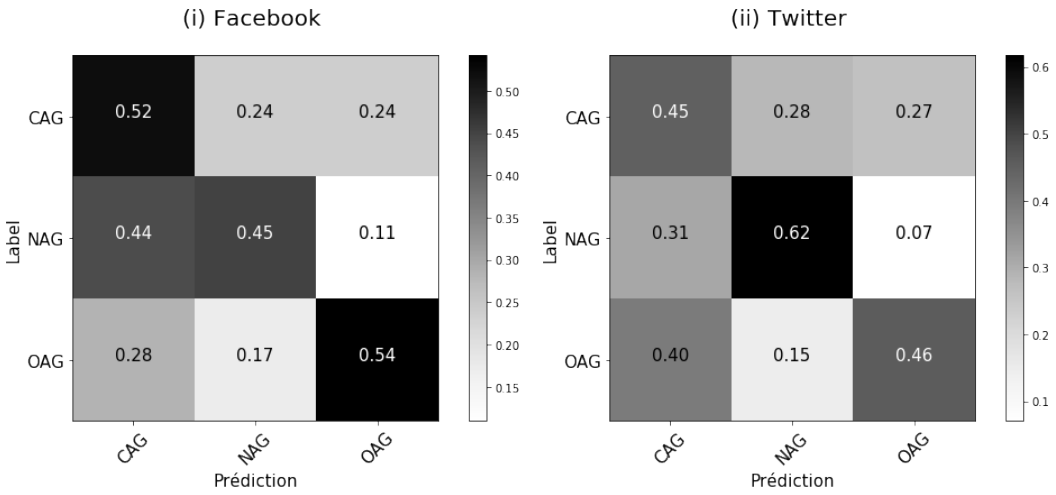
Dépression		Anorexie
2017	2018	
Caractéristiques communes à toutes les collections (suite)		
empath_strength empath_suffering empath_sympathy empath_terrorism empath_ugliness empath_weapon		

Annexe B

TABLEAU 5.2 – Liste des catégories lexicales de Empath

computer ; beauty ; sleep ; timidity ; white collar job ; power ; school ; urban ; sadness ; contentment ; government ; home ; superhero ; leader ; animal ; achievement ; economics ; medieval ; sound ; terrorism ; aggression ; furniture ; friends ; weakness ; leisure ; emotional ; cold ; deception ; rage ; law ; irritability ; weapon ; stealing ; messaging ; payment ; work ; pet ; zest ; toy ; valuable ; office ; ugliness ; ridicule ; domestic work ; affection ; military ; appearance ; shape and size ; dominant heirarchical ; negotiate ; vacation ; science ; hygiene ; cooking ; programming ; college ; youth ; phone ; meeting ; restaurant ; technology ; fashion ; hearing ; play ; tourism ; sympathy ; air travel ; order ; pain ; body ; driving ; beach ; money ; confusion ; legend ; ancient ; competing ; philosophy ; envy ; warmth ; vehicle ; love ; party ; fire ; giving ; weather ; disgust ; fun ; writing ; politics ; speaking ; negative emotion ; farming ; help ; independence ; exotic ; anger ; medical emergency ; business ; suffering ; sexual ; exasperation ; alcohol ; death ; strength ; healing ; torment ; morning ; wealthy ; cheerfulness ; traveling ; music ; noise ; heroic ; real estate ; movement ; trust ; dance ; anonymity ; magic ; banking ; breaking ; water ; occupation ; horror ; ocean ; art ; hate ; rural ; swearing terms ; joy ; sailing ; pride ; poor ; shopping ; night ; gain ; car ; journalism ; celebration ; monster ; disappointment ; dominant personality ; eating ; religion ; childish ; tool ; clothing ; dispute ; fear ; optimism ; sports ; nervousness ; hipster ; cleaning ; kill ; swimming ; liquid ; feminine ; fight ; positive emotion ; crime ; anticipation ; fabric ; communication ; shame ; smell ; prison ; listen ; injury ; health ; children ; masculine ; exercise ; wedding ; divine ; royalty ; violence ; blue collar job ; politeness ; internet ; reading ; family ; plant ; war ; hiking ; social media ; attractive ; lust ; musical ; neglect ; surprise ; ship ; worship.

Annexe C



Matrice de confusion du modèle (Mod_CNN_LSTM) sur les collections de test de TRAC 2018 : (i) Facebook et (ii) Twitter.