Indexation de bout-en-bout dans les bibliothèques numériques scientifiques

Ygor Gallina

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

THÈSE DE DOCTORAT DE

NANTES UNIVERSITE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Ygor GALLINA

Indexation de bout-en-bout dans les bibliothèques numériques scientifiques

Thèse présentée et soutenue à Nantes, le 28 mars 2022
Unité de recherche : Laboratoire des Sciences du Numériques de Nantes (LS2N)

Rapporteurs avant soutenance :

Josiane MOTHE	Professeure, Université de Toulouse (IRIT)
Patrick PAROUBEK	Ingénieur de recherche au CNRS, Université de Paris-Saclay (LISN)

Composition du Jury :

Président :	Richard DUFOUR	Professeur, Nantes Université (LS2N)
Examinateurs :	Josiane MOTHE	Professeure, Université de Toulouse (IRIT)
	Patrick PAROUBEK	Ingénieur de recherche au CNRS, Université de Paris-Saclay (LISN)
	Lorraine GOEURIOT	Maître de conférences, Université Grenoble Alpes (LIG - IUT 1)
	Richard DUFOUR	Professeur, Nantes Université (LS2N)
Dir. de thèse :	Béatrice DAILLE	Professeure, Nantes Université (LS2N)
Co-encadrant :	Florian BOUDIN	Maître de conférences, Nantes Université (LS2N)

Remerciements

Je tiens en tout premier lieu à remercier Florian Boudin pour son encadrement et son soutien indéfectible et sans relâche à chaque étape de cette thèse. Je remercie également Béatrice Daille pour avoir dirigé ma thèse ainsi que pour son aide lors de la rédaction de ce manuscrit.

Je remercie Josiane Mothe et Patrick Paroubek d'avoir accepté d'être relecteur·ices de ma thèse, ainsi que Lorraine Goeuriot et Richard Dufour de faire partie de mon jury. Je remercie aussi les membres de mon CSI, Vincent Claveau et (encore) Josiane Mothe de m'avoir suivie pendant ces trois (et demie) années.

Je remercie les permanent·es de l'équipe TALN ainsi que les membres de l'administration du LS2N avec qui j'ai partagé le quotidien du laboratoire durant ces trois années et demie. Je remercie mes collègues doctorant·es de l'équipe TALN pour nos discussions (plus politiques que scientifiques) et nos parties de GeoGuessr endiablées : Mérième, Martin, Victor, Adrien. Ainsi que les doctorant·es dont j'ai croisé la route et les stagiaires de passage : Esther, Timothée, Oumaima, Mathieu, Rémi, Reda, Fawzi, Kadi, ...

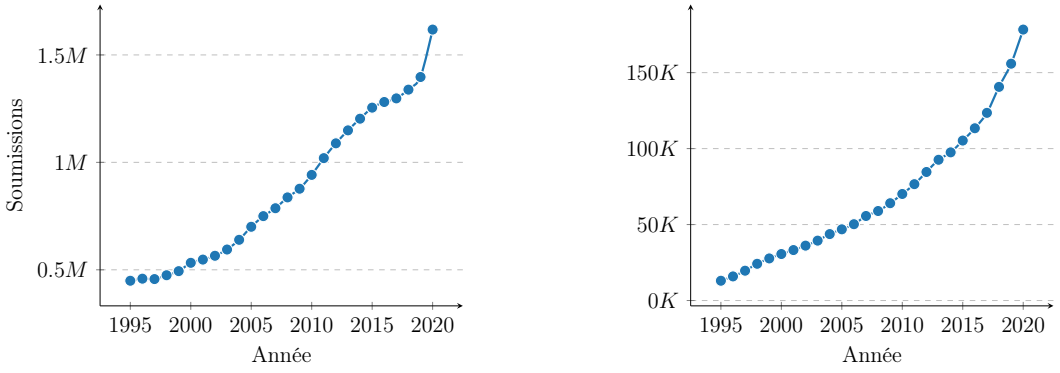
Je remercie ma mère pour son soutien et ses nombreuses relectures de ce manuscrit. Je remercie aussi mes amis : Camille, qui m'a supportée de loin, ainsi que Dewi et Elzéard pour leur soutien durant ces trois années et demie que l'on a rythmées de gavottes des montagnes, de kost ar c'hoad et de sauts basque.

Table des matières

Chapitre 1

Introduction

Les documents scientifiques qui reflètent l'état des connaissances des domaines des sciences sont stockés dans des bibliothèques numériques scientifiques. Pour pouvoir rechercher ces documents, il faut qu'ils soient au préalable indexés par leur contenu, que ce soit par l'entièreté du texte ou par des mots-clés. Les mots-clés représentent les concepts les plus importants d'un document et servent de condensateur textuel, c'est-à-dire qu'ils sont « une expression du texte à la fois réduite du point de vue de la forme et synthétique du point de vue de son "sens" » (Amar, 1997). Ils sont le plus souvent utilisés directement pour indexer des documents mais ils servent aussi à la détection d'opinion (Berend, 2011), à la catégorisation de texte (Hulth and Megyesi, 2006) ou encore à l'élaboration automatique de résumés (Zhang et al., 2004).



(a) PubMed (b) ArXiv

Figure 1.1 – Nombre de nouveaux documents par an dans deux bibliothèques numériques scientifiques.

Le nombre de documents que contiennent les bibliothèques numériques scientifiques ne cesse d'augmenter. La figure 1.1 illustre la croissance exponentielle du nombre de documents déposés chaque année dans ces bibliothèques. À titre d'exemple, plus de 180 000 documents ont été déposés dans la bibliothèque ArXiV en 2020 contre seulement 100 000 en 2015. [1] Ainsi, le nombre d'articles retournés pour une requête précise augmente avec ce nombre croissant de documents et rend plus laborieuse la recherche documentaire. Des moteurs de recherche dédiés facilitent néanmoins cette recherche de documents scientifiques, comme Google Scholar qui est le plus populaire ou encore Microsoft Academics

1. Les données de ArXiV et de PubMed proviennent respectivement de https://arxiv.org/stats/get_monthly_submissions et https://pubmed.ncbi.nlm.nih.gov/?term=all[sb].

et SemanticScholar. Ces derniers permettent de filtrer une partie des documents grâce à l'identification automatique de « sujets » dans les documents.

L'annotation manuelle, réalisée par des documentalistes, atteint ses limites lorsque la masse de documents à traiter est trop importante, d'une part pour être absorbée par les experts disponibles, d'autre part en termes de coût. C'est pourquoi, l'indexation de grandes quantités de documents par mots-clés est aujourd'hui semi- (Mork et al., 2013) ou totalement automatique (Cuxac and Thouvenin, 2017). La production automatique de mots-clés pour un document est une thématique à part entière qui mobilise plusieurs communautés scientifiques dont celle du traitement automatique des langues qui s'y intéresse depuis les années soixante-dix. Les travaux pionniers de Karen Spärck Jones permettent d'identifier les mots importants d'un document grâce à l'introduction de la mesure de TF×IDF (Jones, 1972). La production de mots-clés pour un document consiste d'abord à identifier les concepts importants, puis à ne retenir qu'un certain nombre de ces concepts selon divers critères et enfin à choisir une dénomination de ces concepts parmi toutes les formes linguistiques possibles. Ces étapes nécessitent une expertise tant sur le domaine du document que dans la pratique d'indexation documentaire. En effet l'identification des concepts repose sur une compréhension du document et le choix de leurs dénominations nécessite de respecter certaines caractéristiques linguistiques. De plus, l'ensemble des mots-clés d'un document doit répondre à des contraintes globales : il doit être complet, c-à-d. qu'il doit couvrir le maximum de concepts importants, et être minimal, c-à-d. que leurs sens doivent se recouvrir le moins possible. Les premières méthodes de production automatique de mots-clés sont extractives, ce qui signifie que les mots-clés sont identifiés à l'intérieur du document. Elles fonctionnent en chaîne de traitement et reposent sur des représentations statistiques ou des graphes pour identifier les mots les plus importants grâce à leur fréquence, leur position ou leur centralité dans le document.

Les méthodes présentées par la communauté scientifique sont nombreuses mais elles sont difficilement comparables car la plupart sont évaluées sur des données différentes, prétraitées différemment et selon des protocoles évaluatifs différents. Ce manque de comparabilité ne permet pas d'orienter efficacement la recherche dans le domaine de la production automatique de mots-clés. De plus, ce phénomène de faible comparabilité est exacerbé par le développement de méthodes neuronales de bout-en-bout. Celles-ci ont la capacité de produire une catégorie de mots-clés qui n'avait jusqu'alors reçu que peu d'attention : les mots-clés absents. Ces méthodes, contrairement aux méthodes extractives en chaîne de traitement, peuvent s'abstraire du document et ne sont plus limités aux seules unités textuelles apparaissant dans le document traité. Ces méthodes de bout-en-bout exploitent l'architecture neuronale encodeur-décodeur qui requiert de grandes quantités de données annotées pour être entraînées. Malheureusement, il n'existe qu'un seul grand jeu de données permettant cet entraînement ce qui limite l'analyse et la comparaison des méthodes de bout-en-bout. En plus de cette faible comparabilité, la méthode d'évaluation usuelle compare les mots-clés prédits à une référence par correspondance exacte, ce qui résulte en une évaluation pessimiste des performances des méthodes de production automatique de mots-clés. Cette méthode d'évaluation, intrinsèque, ne considère pas l'utilité des mots-clés

dans des tâches applicatives, comme la recherche d'information par exemple, alors que leur principal intérêt réside justement dans le fait d'être utilisé pour des tâches applicatives.

Ainsi, les objectifs de nos travaux sont triples. Le premier est de démontrer la validité des méthodes de bout-en-bout en les confrontant à différents genres de documents. Jusqu'à présent ces méthodes n'ont été évaluées que sur un seul jeu de données. Cet objectif implique la construction de nouvelles ressources contenant assez de documents annotés pour permettre l'apprentissage de ces modèles neuronaux profonds. Le second objectif est de comprendre la stagnation en termes de performances des méthodes de production de mots-clés, qu'elles soient statistiques, fondée sur les graphes ou neuronales. Ceci implique de comparer ces méthodes et ainsi de les évaluer de manière unifiée. Le troisième objectif est de quantifier l'efficacité des méthodes de production de mots-clés pour une tâche de recherche d'information, et ainsi plus généralement de mesurer l'utilité des mots-clés produits automatiquement dans une tâche applicative.

Pour atteindre ces objectifs, nous faisons quatre hypothèses, présentées ci-dessous. Notre première hypothèse concerne les méthodes de bout-en-bout génératives et la qualité des mots-clés de référence. Elle porte sur la qualité de l'annotation des mots-clés de référence qui devrait influer sur les performances des modèles appris. La disponibilité d'un seul jeu de données de grande taille ne permet pas de répondre à ce questionnement. Quelques travaux s'intéressent à entraîner ces modèles avec plus de données de manière non supervisée (Ye and Wang, 2018) mais à notre connaissance aucun ne s'intéresse à la qualité des mots-clés de référence. La création d'un autre jeu de données de taille comparable à l'existant mais contenant des mots-clés de meilleure qualité permettrait d'étudier la capacité de généralisation de ces modèles. Nous faisons l'hypothèse qu'il est possible de construire automatiquement un jeu de données contenant des mots-clés se rapprochant d'annotations professionnelles et ainsi de consolider les résultats des méthodes neuronales.
Notre deuxième hypothèse concerne la comparaison des performances des méthodes proposées par la communauté scientifique. Les performances de ces méthodes ne peuvent actuellement pas être comparées car elles sont évaluées suivant des cadres expérimentaux différents, et ainsi, ne permettent pas d'avoir une vision d'ensemble des performances de la tâche de production automatique de mots-clés. Nous faisons l'hypothèse qu'il est possible de proposer un cadre expérimental unifié pour reproduire les expériences et ainsi pouvoir évaluer et comparer les méthodes état-de-l'art.
Notre troisième hypothèse concerne l'évaluation des méthodes de production automatique de mots-clés. Le seul cadre d'évaluation intrinsèque ne permet pas de mesurer efficacement les performances et l'utilité des méthodes proposées. Quelques travaux s'intéressent néanmoins à améliorer l'évaluation intrinsèque en assouplissant la comparaison avec la référence (Zesch and Gurevych, 2009) ou en l'étendant avec des variantes (Kim et al., 2010; Chan et al., 2019). Nous faisons l'hypothèse qu'il est possible de définir un protocole expérimental dans le cadre de la recherche d'information pour évaluer les méthodes de production automatiques de mots-clés.
Notre dernière hypothèse concerne les mots-clés absents que les méthodes de bout-en-bout

permettent dorénavant de produire. Cette catégorie de mots-clés permet, dans le cadre de leur indexation, d'enrichir les documents en y ajoutant de nouveaux mots. Dans le cadre de la recherche de documents, cet enrichissement permet d'augmenter la couverture des documents retournés. Ainsi, nous faisons l'hypothèse que ces mots-clés, qui n'apparaissent pas dans les documents, ont un plus grand impact sur l'amélioration des performances des systèmes de recherche d'information, que les mots-clés présents dans les documents.

Nous présentons tout d'abord dans le chapitre 2 les concepts importants de cette thèse que sont l'indexation et les mots-clés, ainsi qu'un état de l'art des méthodes automatiques d'extraction de mots-clés. Le chapitre 3 présente un état de l'art des méthodes de production automatique de mots-clés de bout-en-bout. Le chapitre 4 présente le cadre expérimental de la production automatique de mots-clés. Nous y présentons les jeux de données utilisés, ainsi que le processus d'évaluation des systèmes. Le chapitre 5 présente le jeu de données KPTimes que nous avons créé et grâce auquel nous évaluons la généralisation des méthodes de production automatique de mots-clés. Dans le chapitre 6 nous présentons une évaluation comparative des méthodes de l'état de l'art sur différents jeux de données. Puis, nous nous intéressons dans le chapitre 7 à l'impact des mots-clés sur l'indexation de documents scientifiques dans le cadre applicatif de la recherche d'information. Enfin nous présentons nos conclusions et perspectives dans le chapitre 8.

Chapitre 2

Concepts et méthodes de base

Ce chapitre présente les concepts qui sont importants pour comprendre le contexte dans lequel s'inscrit cette thèse. Nous décrivons tout d'abord l'indexation de documents scientifiques, qu'elle soit manuelle ou automatique. Nous nous intéressons ensuite aux mots-clés et à leurs principales caractéristiques. Enfin, nous présentons un état de l'art des méthodes d'extraction de mots-clés en chaîne de traitement, en commençant par décrire les trois étapes de cette chaîne : l'identification de candidats, leur pondération, puis la sélection d'un ensemble de mots-clés parmi ces candidats. Nous présenterons dans le chapitre suivant un état de l'art des méthodes neuronales de bout-en-bout.

2.1 Indexation de documents scientifiques

L'indexation est un processus qui vise à identifier les éléments notables d'un document dans le but de le caractériser (Khemiri and Sidhom, 2020). L'indexation par mots-clés, ou association de mots-clés à des documents, est à l'origine un processus manuel, effectué par des indexeurs professionnels ou des bibliothécaires formés à cette problématique. Dans les bibliothèques, les documents sont généralement associés à des mots-clés qui proviennent de vocabulaires contrôlés. Par exemple, les bibliothèques universitaires indexent leurs documents grâce au langage documentaire RAMEAU (Centre National RAMEAU, 2017) qui permet de décrire les sujets des documents grâce à des descripteurs. Dans ce langage documentaire, un document intitulé « Les événements de mai 68 racontés par un étudiant » sera indexé avec les descripteurs suivants : France – 1968 (Journées de mai) – Récits personnels ; ou encore le document « Les conditions de travail des enseignants en Bretagne » sera indexé de la manière suivante : Enseignants – France – Bretagne (France) – Conditions de travail. [1]

2.1.1 Indexation manuelle

L'indexation manuelle par mots-clés, appelée aussi annotation manuelle de documents en mots-clés, peut s'effectuer de manière contrôlée ou non contrôlée. De manière contrôlée, les mots-clés sont à choisir dans un référentiel (ontologie, thésaurus, base de données

1. Exemples extraits de https://rameau.bnf.fr/sites/default/files/formation/pdf/ex2corr.pdf

Agrégats de mots-clés validés sémantiquement : Pour de nouveaux services d'accès à l'information sur internet (id : `sciencesInfo_10-0065090_tei`)

A l'heure du web social, nous présentons une solution destinée à définir de nouveaux services tels que la construction automatique et dynamique de communautés d'utilisateurs : l'agrégation de mots-clés. Ces agrégats de mots-clés sont issus des recherches antérieures des utilisateurs réalisées au travers d'un moteur de recherche. Nous présentons la démarche que nous avons suivie pour obtenir un algorithme de regroupement des mots-clés provenant de fichiers de traçage (log) ; nous illustrons cet algorithme au travers de son application au fichier de traçage du moteur de recherche aol.com. A des fins d'évaluation et de validation, nous proposons de comparer les résultats obtenus par le moteur de recherche à partir des agrégats de mots-clés ainsi créés et de définir un coefficient de cohérence sémantique de ces agrégats. Nous mesurons dans une expérimentation la perte de cohérence sémantique liée à l'augmentation de la taille des agrégats. L'intérêt de notre approche réside dans le fait qu'elle peut être considérée comme une brique de base pour un grand nombre de systèmes « communautaires » et ainsi exploitée pour offrir encore plus de services à l'usager.

Mots-clés de référence : Classe, Fichier log, Agrégat, Mot clé, Traitement de la requête, Communauté virtuelle, Réseau social, Recherche information, Algorithme

FIGURE 2.1 – Exemple de notice scientifique des bases bibliographiques Pascal et Francis. Les mots communs entre le document et les mots-clés sont colorés.

terminologiques, etc.). De manière non contrôlée, le choix des mots-clés est à la discrétion de l'annotateur. Pour illustrer cette indexation par mots-clés, nous présentons dans la figure 2.1 un exemple de notice scientifique annotée en mots-clés par des indexeurs professionnels.

L'annotation contrôlée permet d'assurer une cohérence dans le choix des termes mais limite le nombre de concepts. Elle nécessite aussi une connaissance experte du référentiel utilisé, par exemple le MeSH dans le domaine médical, c'est pourquoi des indexeurs professionnels sont formés à leur utilisation. Le MeSH contient 25 186 termes [2] organisés hiérarchiquement avec quatre niveaux de profondeur en moyenne. Pour faciliter cette annotation contrôlée, des outils d'annotation semi-automatique, tels que le Medical Text Indexer (Mork et al., 2013) pour PubMed, suggèrent aux indexeurs les mots-clés du référentiel qui apparaissent dans les documents. Les indexeurs procèdent ensuite à un examen manuel des mots-clés suggérés pour valider ou ajouter des mots-clés du référentiel qui n'ont pas été détectés par ces outils. En contrepartie de la qualité de ces référentiels, leur mise à jour et leur construction sont de lourds processus qui doivent toujours prendre en compte l'intégralité du référentiel pour garantir sa cohérence.

L'indexation non contrôlée, contrairement à l'indexation contrôlée, n'est soumise à aucune contrainte. Elle permet une annotation réalisable sans connaissances préalables mais impacte négativement la cohérence de l'annotation d'un document à l'autre. Cette incohérence est montrée dans la figure 2.2 qui regroupe les variantes du concept de *neural network* dans des documents scientifiques annotés par leurs auteurs. L'indexation non contrôlée permet aussi, contrairement à l'indexation contrôlée, d'indexer des concepts émergeant et n'est pas limitée aux termes déjà identifiés par un référentiel. Cette indexation non contrôlée est principalement utilisée dans les bibliothèques numériques scientifiques, car les documents qui comportent des mots-clés sont pour la plupart annotés par leurs auteurs lors de l'écriture ou de la soumission des articles.

2. https://www.nlm.nih.gov/databases/download/mesh.html

Variantes racinisées	Fréquence	Variantes racinisées (*suite*)	Fréquence
neural network	5612	nn	7
artifici neural network	2083	artifici neural net	6
artifici neural network (ann)	147	nn neural network	6
neural net	138	artif neural network	4
neural network model	79	neural networks.	3
neural model	70	ann (artifici neural net)	2
neural network (nn)	36	ann (artifici neural networks)	2
artifici neural network (anns)	34	artifici neural network (anni)	1
ann artifici neural network	25	ann : artifici neural network	1
neural network (nns)	24	arti ?ci neural network	1
neural-network	9	artifici neural networks. cad	1
the neural network	8	ann ann artifici neural network	1
artifici neural network model	7	nn nn neural network	1

FIGURE 2.2 – Variantes racinisées du concept de *neural network* trouvées dans les mots-clés de référence de KP20k.

L'annotation en mots-clés, qu'elle soit contrôlée ou non, est généralement effectuée par des auteurs, des lecteurs ou des indexeurs professionnels.
Les **auteurs** fournissent des mots-clés pour les documents qu'ils ont écrits, ils ont donc une connaissance experte du domaine et du contenu du document. Les mots-clés qu'ils choisissent décrivent les concepts importants de leur point de vue et peuvent omettre certains concepts abordés. De plus, le choix des mots-clés peut être biaisé par les thématiques populaires du moment dans le but d'augmenter la visibilité de l'article. L'annotation par les auteurs est très peu cohérente car il n'y a pas de guide d'annotation, et chaque document est annoté par une personne différente. La figure 2.2 présente des variantes du concept de *neural network* (« réseau de neurone » en français) annotés par des auteurs.
Les **lecteurs**, quant à eux, ne sont pas des experts de l'annotation mais peuvent être experts du domaine du document. Au contraire des auteurs, leur but n'est pas la visibilité du document mais plutôt l'identification des concepts des documents qui leur sont personnellement utiles dans leur recherche documentaire. Les annotations lecteurs – dans le cadre de documents scientifiques – proviennent généralement de plateformes de partage de bibliographies dans lesquels les utilisateurs peuvent associer des mots-clés à des documents. Dans un cadre de création de jeux de données pour la production automatique de mots-clés, les données des utilisateurs sont compilées et filtrées pour obtenir un ensemble de mots-clés associé à chaque document retenu. Cette annotation lecteur permet, par exemple, d'offrir une annotation alternative à une annotation déjà présente ou tout simplement d'obtenir une annotation en mots-clés moins coûteuse qu'une annotation professionnelle.
Les **indexeurs professionnels**, pour leur part, sont formés à l'indexation et à l'utilisation de langages documentaires. Ils peuvent avoir une expertise dans le domaine des documents à annoter et ont pour objectif d'affecter des mots-clés qui facilitent la recherche documentaire pour les utilisateurs.

Pour illustrer la différence d'annotation entre les auteurs et les indexeurs profession-

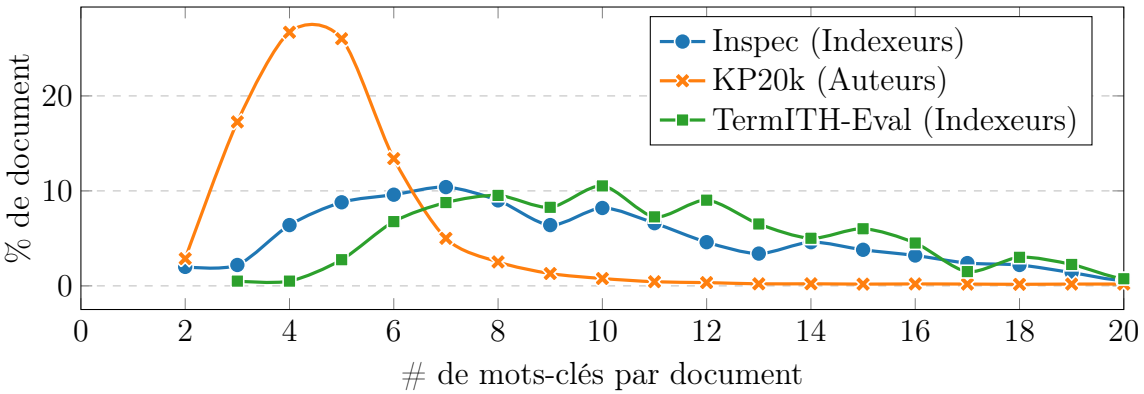


FIGURE 2.3 – Nombre de mots-clés par document annoté par différentes catégories d'annotateurs.

nels, nous comparons le nombre de mots-clés assigné aux documents par ces deux types d'annotateurs. Ainsi, la figure 2.3 présente la fréquence de documents par nombre de mots-clés pour trois jeux de données de notices scientifiques : Inspec et KP20k en anglais ; TermITH-Eval en français. La différence entre l'annotation indexeur et l'annotation auteur est flagrante. En effet, les auteurs assignent le plus souvent cinq mots-clés par document, ce qui correspond au nombre maximal de mots-clés autorisés par les éditeurs de documents scientifiques, alors que les indexeurs, qui ne sont pas contraints par un seuil maximal, annotent en majorité de 6 à 10 mots-clés par document, sans différence entre le français et l'anglais.

2.1.2 Indexation automatique par mots-clés

L'indexation automatique consiste à caractériser des documents de manière automatique, c'est-à-dire à choisir et à pondérer les descripteurs d'un document de manière automatique. L'indexation plein texte est un type d'indexation automatique qui considère chaque mot du document comme un descripteur potentiel, puis lui attribue un poids selon un schéma de pondération tel que TF×IDF.

Les techniques d'indexation automatique ont été développées pour simplifier et accélérer le travail d'indexation jusque-là manuel. Ce travail nécessite la disponibilité d'experts ainsi que des budgets conséquents : l'annotation manuelle d'un article de PubMed coûte une dizaine de dollars [3] ; en 2020, 1,5 million d'articles ont été ajoutés à PubMed ce qui représente un budget de 10,5 millions de dollars pour cette seule année. Ce processus est aussi coûteux en temps : il faut compter entre 2 et 3 mois entre la soumission d'un document et son indexation. Ce délai d'attente découle de la masse de documents à indexer.

Nous nous intéressons ici à l'indexation automatique par mots-clés que nous considérons comme un type d'indexation libre. Et plus particulièrement, nous nous intéressons à la production automatique de mots-clés. Les mots-clés sont des unités textuelles qui

3. lhncbc.nlm.nih.gov/ii/information/about.html

représentent les sujets importants d'un document. Nous les présenterons en détail dans la section 2.2. Les mots-clés ont de multiples intérêts pour l'indexation automatique de documents : ils peuvent aider à la création de thésaurus (Kosovac et al., 2002) ou autre référentiel ; ils peuvent aussi aider à la création de résumés automatiques (Litvak and Last, 2008; Qazvinian et al., 2010). Par ailleurs, ils peuvent enrichir l'indexation plein texte ou encore être utilisés pour de la recherche à facette (Gutwin et al., 1999).

La tâche qui consiste à associer automatiquement des mots-clés à des documents est généralement nommée « extraction de mots-clés » (*keyphrase extraction*) (Hasan and Ng, 2014; Meng et al., 2017). La grande majorité des méthodes de production automatique de mots-clés proposée avant 2017 sont extractives, c'est-à-dire qu'elles produisent des mots-clés présents dans le document. En 2017, Meng et al. (2017) introduit une méthode supervisée générative qui génère des mots-clés mot-à-mot à partir d'un vocabulaire. Cette méthode permet donc non seulement de produire des mots-clés présents mais aussi des mots-clés absents du document.

Le terme « extraction de mots-clés » est ambigu : il peut désigner la seule production de mots-clés présents, ou bien la production de mots-clés indifféremment présents ou absents. Dans ce travail de thèse, nous réservons le terme d'extraction de mots-clés à la seule extraction de mots-clés apparaissant dans le document. Pour l'affectation de mots-clés à un document, qu'ils soient présents ou absents du document, nous emploierons « assignation de mots-clés » si les mots-clés proviennent d'un vocabulaire contrôlé et « génération de mots-clés » si les mots-clés sont générés par des modèles supervisés ou semi-supervisés. Le terme « production de mots-clés » désignera indifféremment l'extraction, l'assignation ou la génération de mots-clés.

2.2 Définition et caractéristiques des mots-clés

Dans cette section nous examinons deux propriétés des mots-clés : les catégories grammaticales de leurs composants et leur longueur. Nous illustrons ces propriétés à l'aide de deux jeux de données : KP20k pour l'anglais et TermITH-Eval pour le français.

Dans la littérature, « mots-clés » et « termes-clés » sont utilisés de manière interchangeable pour désigner les concepts importants d'un document. [4] Ces deux appellations peuvent parfois être utilisées afin de différencier les mots-clés comprenant plusieurs mots (termes-clés) des unigrammes (mots-clés) mais cette utilisation n'est pas systématique. Dans ce travail de thèse, nous choisissons d'employer « mot-clé » pour désigner ces concepts importants sans rapport avec le nombre de mots qui les composent, ni le fait que les « mots-clés » soient des termes (d'un point de vue terminologique). [5]

L'indexation des documents se fait toujours en leur associant des ensembles de mots-

4. « descripteurs » peut aussi être utilisé dans le contexte de la recherche d'informations.

5. Les travaux de (Bougouin, 2015) portent sur les domaines de spécialité d'où son utilisation de « terme-clé ».

clés. Ces ensembles doivent respecter les propriétés de non-redondance et de couverture, c'est-à-dire que les mots-clés qui les composent doivent être sémantiquement disjoints, et couvrir le plus de concepts importants du document (Firoozeh et al., 2020). Au niveau d'une collection de documents, les mots-clés peuvent être plus ou moins cohérents, c'est-à-dire qu'un concept est représenté par un nombre plus ou moins grand de variantes. L'exemple du concept de *neural network* dans la figure 2.2 met en lumière ce phénomène.

2.2.1 Nature linguistique des mots-clés

Dans cette section, nous définissons les caractéristiques linguistiques des mots-clés. Selon l'étude de Hulth (2003) les mots-clés sont majoritairement des noms et des expressions nominales, et sont donc composés de noms et d'adjectifs.

Fréquence	Patron			Exemple
21,2	NOUN			graphs
17,2	NOUN	NOUN		similarity measure
14,7	ADJ	NOUN		empirical study
4,5	VERB			denoising
4,1	ADJ	NOUN	NOUN	ant colony optimization

(a) Mots-clés anglais (KP20k)

Fréquence	Patron			Exemple
31,7	NOUN			internet
19,7	NOUN	ADJ		paléolithique moyen
9,8	ADJ			historique
4,5	PROPN			europe
4,2	NOUN	ADP	NOUN	langue de spécialité

(b) Mots-clés français (TermITH-Eval)

FIGURE 2.4 – Patrons syntaxiques de mots-clés ordonnés par fréquence. Les mots-clés ont été étiquetés à l'aide de la bibliothèque `spacy` (version 3.1 des modèles français et anglais). Le jeu d'étiquettes utilisés est l'Universal Dependencies POS-tags (Petrov et al., 2012).

Pour confirmer ce résultat sur les différents jeux de données disponibles, nous avons calculé la fréquence des patrons morphosyntaxiques des mots-clés sur le jeu de données anglais KP20k et le jeu de données français TermITH-Eval. La figure 2.4 présente les 5 patrons morphosyntaxiques les plus fréquents. Ces 5 patrons couvrent respectivement 62 % et 70 % des mots-clés de KP20k et de TermITH-Eval. Dans les deux langues, quatre des cinq patrons sont exclusivement composés de noms et d'adjectifs, ce sont donc des syntagmes nominaux. En anglais, 4,5 % des mots-clés sont des verbes ; en français les noms propres (assimilables à des noms) représentent 4,5 % des mots-clés. La faible proportion du patron `NOUN ADP NOUN` en français est surprenante compte tenu de sa prépondérance dans les domaines de spécialités (Daille, 2017).

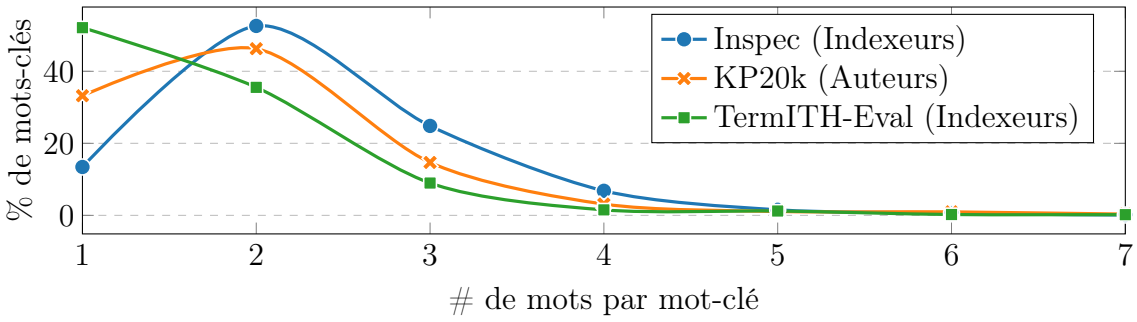


FIGURE 2.5 – Nombre de constituants par mot-clé dans les ensembles de test de différents jeux de données.

Les mots-clés doivent être précis et concis : ils sont donc généralement assez courts. La figure 2.5 présente le nombre moyen de constituants par mot-clé dans deux jeux de données annotés par des indexeurs professionnels (Inspec en anglais et TermITH-Eval en français) et un par des auteurs (KP20k en anglais). Ces jeux de données sont présentés en détail dans la section 4. Nous observons que les mots-clés sont en très grande majorité composés de 1 à 3 mots, $\simeq$90 % pour les trois jeux de données. La seule différence notable entre ces trois jeux de données est le nombre de mots-clés unigrammes qui sont beaucoup plus nombreux en français qu'en anglais. Notons que quelques mots-clés contiennent plus de 10 constituants. Ce sont généralement soit des expansions d'acronymes « samovar (système d'analyse et de modélisation des validations et des automobiles renault) », soit des mots-clés exprimant des notions très spécifiques comme « iterative regularized least-mean mixed-norm image restoration », soit des noms d'entités comme les noms de molécules « benzènesulfonique acide(méthyl-4) [méthyl-«5p» isoxazolyl-«3p»] amide ».

Pour aider les auteurs à choisir les mots-clés de leurs articles, Gbur and Trumbo (1995) donnent des recommandations pour l'anglais. Par exemple, ils recommandent de ne pas répéter les mots-clés des titres, de ne pas choisir de mots-clés trop communs (« regression » dans le domaine des statistiques) et de choisir des syntagmes nominaux simples et spécifiques qui évitent les composés syntagmatiques avec groupe prépositionnel (« reliability » plutôt que « theory of reliability ») etc.

2.2.2 Mots-clés présents et mots-clés absents

La notion d'absence d'un mot-clé a été introduite et formalisée par Meng et al. (2017) dans les termes suivants : « [...] nous dénotons les mots-clés qui ne correspondent à aucune sous-séquence continue du texte source comme des mots-clés absents, et ceux qui correspondent à une partie du texte comme des mots-clés présents ». [6] Cette définition est implémentée en cherchant si la séquence de mots du mot-clé apparaît dans le même

6. « *[...] we denote phrases that do not match any contiguous subsequence of source text as absent keyphrases, and the ones that fully match a part of the text as present keyphrases.* »

ordre que dans la séquence de mots du texte source. [7] Ce découpage permet de différencier les mots-clés pouvant être extraits du document (mots-clés présents) de ceux devant être générés (mots-clés absents). Cette différenciation est généralement utilisée pour filtrer la référence et pour évaluer une méthode sur sa seule capacité à extraire ou à générer des mots-clés. Les méthodes extractives ont historiquement été évaluées à l'aide de la référence entière. Aujourd'hui il est commun d'évaluer séparément les mots-clés présents et les mots-clés absents (Meng et al., 2017; Sun et al., 2019).

2.3 Méthodes en chaîne de traitement

Dans cette section, nous présentons les méthodes d'extraction automatique de mots-clés en chaîne de traitement. Ces méthodes sont dites en « chaîne de traitement » car elles s'exécutent en trois étapes :

1. l'identification d'unités textuelles que l'on va considérer comme candidates à être des mots-clés ;
2. la pondération de ces mots-clés candidats, à l'aide de différentes méthodes : statistiques, de classification, utilisant des graphes, etc. ;
3. la sélection d'un sous-ensemble de candidats qui représente le document.

La figure 2.6 illustre le déroulement en trois étapes des méthodes en chaîne de traitement.

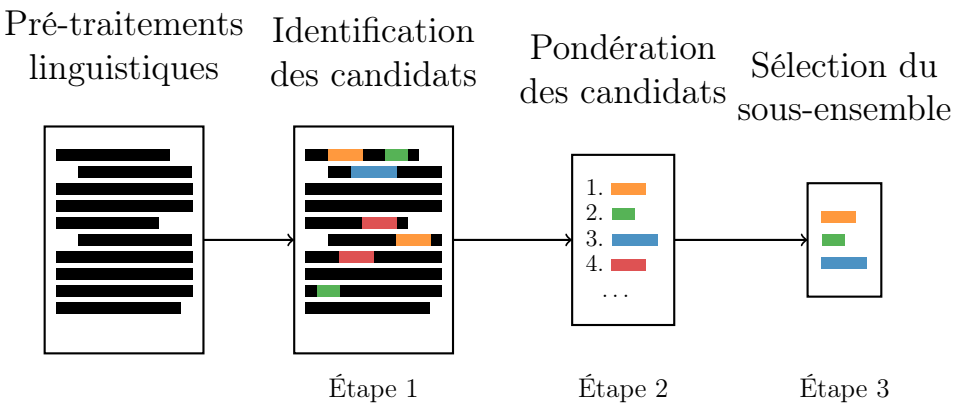


FIGURE 2.6 – Étapes principales des méthodes d'extraction de mots-clés en chaîne de traitement.

2.3.1 Identification des mots-clés candidats

L'identification des mots-clés candidats est la première étape de l'extraction de mots-clés. Elle consiste à identifier certaines unités textuelles du document qui peuvent être des mots-clés. Cette identification met en œuvre des heuristiques exploitant certaines

7. github.com/memray/seq2seq-keyphrase/keyphrase/keyphrase_utils.py#L96

caractéristiques des mots-clés (fréquence, position, patron morphosyntaxique, ...). Le choix de la méthode de sélection des mots-clés candidats influe grandement sur la tâche d'extraction car elle pose une hypothèse forte sur les propriétés que doivent respecter les mots-clés. Sélectionner un grand nombre de candidats complexifie l'étape suivante de pondération. En sélectionner trop peu risque de limiter les performances de l'extraction de mots-clés en occultant certains mots-clés pertinents. Cette étape doit donc trouver un compromis entre minimiser le bruit (le nombre de candidats non pertinents) et maximiser le rappel (le nombre de candidats pertinents).

Deux méthodes principales sont employées pour la sélection de mots-clés : les n-grammes et les patrons morphosyntaxiques. Les n-grammes (séquences continues de n mots) sélectionnés sont généralement des unigrammes, bigrammes et trigrammes (Witten et al., 1999; Campos et al., 2020). Cette méthode produit un nombre conséquent de mots-clés candidats mais elle garantie une couverture de l'ensemble du document. Les n-grammes candidats sont ensuite filtrés pour éliminer les séquences peu susceptibles d'être des mots-clés, comme celles qui commencent et finissent par des catégories grammaticales fonctionnelles ou encore celles qui ne contiennent ni nom ni adjectif.

Les méthodes les plus populaires sélectionnent des séquences d'étiquettes morphosyntaxiques à l'aide de patrons morphosyntaxiques. Nous avons vu dans la section 2.2.1 que les mots-clés sont en grande majorité composés de noms et d'adjectifs. Le patron générique le plus populaire est `/(NOUN|ADJ)+/` (Mihalcea and Tarau, 2004; Wan and Xiao, 2008a; Bougouin et al., 2013, *inter alia*). Construire des patrons morphosyntaxiques plus précis, décrivant des mots-clés, requiert soit l'expertise linguistique de spécialistes de la langue, soit l'acquisition automatique de patrons à partir d'un ensemble d'apprentissage annoté en mots-clés. Dans cette direction, Hulth (2003) considère comme patrons acceptables tous les patrons apparaissant au moins 10 fois dans les mots-clés des documents d'entraînement. Les outils d'étiquetage morphosyntaxique sont aujourd'hui des outils de traitement linguistique de base pour de nombreuses langues à l'exception de certaines langues peu dotées. En terme de performances, l'étiqueteur morphosyntaxique de l'outil `spacy`[8] atteint une précision de 97 % pour l'anglais et 93 % pour le français. Ces bonnes performances globales cachent néanmoins des problèmes récurrents : en anglais, un problème d'étiquetage des extensions du nom de tête des composés anglais par exemple dans « *ant colony optimization* », *ant* est catégorisé comme adjectif au lieu de nom ; en français, la confusion de noms en verbes dès que la forme du nom est aussi une forme flexionnelle acceptable du verbe, par exemple le nom « défigement » fini par « ent » et est catégorisé comme verbe. Pour le français, par exemple, les mots-clés peuvent contenir des prépositions comme « langue *de* spécialité » ; le patron `(NOUN|ADJ)+` ne permet pas la sélection de ces composés nominaux pourtant très fréquents (Daille, 2017).

La figure 2.7 donne un exemple de candidats sélectionnés par la méthode n-grammes et selon le patron `/(NOUN|ADJ)+/`. Le nombre de candidats identifiés en conservant les n-grammes de 1 à 3 (12 candidats) est très supérieur à ceux identifiés par le patron

8. https://spacy.io/models

`/(NOUN|ADJ)+/` (2 candidats). La plupart des candidats identifiés par la méthode n-gramme n'apporte que peu d'informations quant aux sujets du document.

1-grammes	indexation ; libre ; ensemble ; unités ; descriptives ; utilisés ; connu ; priori
2-grammes	indexation libre ; unités descriptives
3-grammes	ensemble des unités ; connu a priori
Patron `(N|A)+`	indexation libre ; unités descriptives

FIGURE 2.7 – Mots-clés candidats identifiés par la méthode n-grammes et selon le patron `(NOUN|ADJ)+` dans la phrase « *Dans l'indexation libre, l'ensemble des unités descriptives qui peut être utilisé n'est pas connu a priori.* » provenant de Neveol (2005).

2.3.2 Pondération des mots-clés candidats

L'étape de pondération des mots-clés candidats consiste à leur assigner un score évaluant leur potentialité à être un mot-clé. Nous présentons les méthodes de l'état de l'art en les regroupant par catégories : les méthodes statistiques qui utilisent des statistiques descriptives, les méthodes utilisant des graphes pour représenter les documents, les méthodes de classification supervisées et d'autres méthodes ne rentrant pas dans ces catégories.

Méthodes statistiques

La méthode statistique la plus utilisée est le TF×IDF (Jones, 1972). C'est une méthode de référence très populaire dans la plupart des tâches de traitement automatique de la langue. Le TF×IDF est un schéma de pondération qui exploite la fréquence des mots dans une collection de documents. Sa formule est décrite dans l'équation 2.1 avec N le nombre de documents de la collection, $\text{DF}(w)$ le nombre de documents comportant le mot w, $\text{TF}_d(w)$ le nombre d'occurrences du mot w dans le document d. L'idée étant que la fréquence élevée d'un mot ou sa spécificité à un document sont des indicateurs d'importance de ce mot.

$$\text{TF×IDF}(d, w) = \text{TF}_d(w) * log\left(\frac{N}{\text{DF}(w)}\right) \tag{2.1}$$

Outre la fréquence, la position d'un candidat dans le document est un indicateur fort de sa propension à être un mot-clé. La position est utilisée dans plusieurs méthodes (Witten et al., 1999; Nguyen and Kan, 2007; Campos et al., 2020, *inter alia*) que nous décrivons ensuite. C'est en effet au début d'un article scientifique, dans son titre et son résumé, que l'on va mentionner les concepts décrits dans l'article. Dans la tâche de résumé automatique, la méthode de base, à laquelle se comparer, utilise les premières phrases du

document qui sont utilisées comme résumé (Brandow et al., 1995). La méthode « First-Phrases » (Gallina et al., 2020) s'inspire de cette méthode et ordonne les candidats selon leur position dans le document.

La méthode YAKE! (Campos et al., 2020) est une méthode qui calcule le score des candidats à l'aide de plusieurs descripteurs statistiques. L'intérêt de ces descripteurs est qu'ils sont indépendants de la langue et – à l'inverse de TF×IDF – ne nécessitent pas de collection de documents. En plus de la fréquence et de la position, la casse et les coocurrences sont aussi utilisées.

Méthodes de classification

L'extraction de mots-clés peut être reformulée comme une tâche de classification binaire. Chaque candidat est classé comme mot-clé ou non mot-clé. La confiance du classifieur dans sa prédiction peut être utilisée pour donner un score aux candidats et donc pour les classer.

La méthode historique Kea (Witten et al., 1999) combine seulement deux descripteurs pour classifier les mots-clés candidats : le TF×IDF et la position de la première occurrence. Cette méthode bien que simple est aujourd'hui toujours compétitive, elle exploite la fréquence et la position qui sont deux caractéristiques fortement liées à l'importance d'un mot.

Certaines méthodes se concentrent sur l'extraction de mots-clés dans les articles scientifiques et peuvent ainsi tirer parti de leur structure. Dans leurs travaux Nguyen and Kan (2007) remarquent que les mots-clés n'apparaissent pas de manière équiprobable dans toutes les sections des articles scientifiques. Pour prendre en compte cette information, un classifieur détecte l'apparition d'un mot-clé candidat dans 14 types de sections avec une précision de 92 %. Parmi ces 14 types de sections se trouvent : le résumé, l'introduction, la conclusion, les travaux connexes, les références, etc. Cette information est ensuite utilisée comme trait supplémentaire pour classifier les mots-clés candidats.

Méthodes fondées sur les graphes

Les méthodes fondées sur les graphes sont très populaires pour l'extraction de mots-clés. Les graphes permettent de calculer des mesures de centralité des nœuds qui dénotent leur importance. Dans le cadre de l'extraction de mots-clés, le graphe $G = (V, E)$ est composé de noeuds V, qui représentent les mots d'un document, et d'arêtes E, qui représentent les relations de cooccurrence entre les mots selon une fenêtre glissante de n mots.

La méthode pionnière utilisant des graphes est TextRank (Mihalcea and Tarau, 2004). Cette méthode applique l'algorithme PageRank (Page et al., 1999) pour pondérer les noeuds et donc les mots du document. L'idée de cet algorithme est qu'un mot est important s'il cooccurre avec un grand nombre de mots, et si les mots avec lesquels il cooccurre

sont eux aussi importants. Ensuite, les candidats sont pondérés en sommant les scores des mots qui les composent. Cette méthode, bien que précurseure, obtient généralement des performances assez basses. D'autres méthodes ont été présentées pour l'améliorer.

Les méthodes CollabRank (Wan and Xiao, 2008a) [9] et CiteTextRank (Gollapalli and Caragea, 2014) s'inspirent de TextRank et améliorent la représentation sous forme de graphe grâce à des données supplémentaires. CollabRank crée le graphe à partir du document et de documents proches identifiés grâce à un algorithme de regroupement (*clustering*), tandis que CiteTextRank traite les articles scientifiques et ajoute au graphe les contextes de citations de l'article. CiteTextRank semble obtenir de meilleures performances que CollabRank mais nécessite d'avoir accès à des articles scientifiques ainsi qu'à leurs contextes de citation. Malheureusement, ces informations sont peu disponibles et difficilement accessibles.

Les méthodes TopicalPageRank (Liu et al., 2010), TopicRank (Bougouin et al., 2013) et son amélioration MultipartiteRank (Boudin, 2018) améliorent TextRank en regroupant par sujets les mots du document ou les candidats. TopicalPageRank utilise le LDA pour calculer les sujets auxquels appartiennent les mots. Le LDA (Blei et al., 2003) (*Latent Dirichlet Allocation*) est une technique de modélisation en sujets qui s'inscrit dans les techniques de réduction de dimensionalité. Le LDA est un modèle statistique génératif qui permet de représenter un document en une mixture de sujets, à leur tour représentés par une mixture de mots. L'algorithme PageRank est biaisé pour prendre en compte l'apport des mots dans chacun des sujets. Les méthodes TopicRank et MultipartiteRank quand à elles, regroupent les mots-clés candidats sur la base du nombre de mots communs. Cette technique est plus simple que LDA mais ne requiert pas d'entraînement. Ces deux méthodes, contrairement à TextRank et à la majorité des autres méthodes, modélisent les candidats plutôt que les mots et forment un graphe complet en pondérant les arêtes par la distance entre les candidats dans le document.

Parmi les méthodes proposées, certaines modifient TextRank en biaisant l'algorithme PageRank pour prendre en compte une caractéristique importante des mots-clés. Par exemple PositionRank (Florescu and Caragea, 2017) biaise PageRank en fonction des positions des mots dans le document. Ce seul changement apporte un gain de performance important par rapport à TextRank qui montre l'intérêt de cette caractéristique pour l'identification de mots-clés.

Dans la grande majorité des méthodes présentées, la construction du graphe et la pondération des arêtes sont basée sur les relations de cooccurrences. Pour tenter de capturer les relations sémantiques entre les mots, Mothe et al. (2018) étudie l'utilisation de la similarité sémantique pour pondérer les arêtes du graphe à l'aide de plongements de mots statiques. Leurs expériences montrent que l'utilisation des plongements de mots seul ou leur combinaison avec la coocurence n'apportent pas de gain de performances significatif.

9. Parfois nommée ExpandRank.

Autres méthodes

Une des premières méthodes à exploiter les représentations denses de mots pour l'extraction de mots-clés non supervisée est EmbedRank (Bennani-Smires et al., 2018). Cette méthode pondère les mots-clés candidats en fonction de leur distance sémantique au document. Pour calculer cette distance sémantique, le document et les mots-clés candidats sont d'abord représentés sous forme vectorielle grâce à une technique de plongements de phrases (Pagliardini et al., 2018). Enfin, le poids est calculé par la distance cosinus entre le vecteur du document et les vecteurs des candidats.

La méthode TopicCoRank (Bougouin, 2015) est une méthode fondée sur les graphes qui utilise l'ensemble des mots-clés de référence pour pouvoir prédire des mots-clés n'apparaissant pas dans le document. Pour cela, l'ensemble des mots-clés de référence d'un jeu de données sont représentés dans un « graphe du domaine » dans lequel deux mots-clés sont connectés s'il apparaissent dans les mêmes ensembles de référence. Ce graphe du domaine est connecté au graphe du document [10] par les mots-clés communs aux deux graphes. De manière similaire aux autres méthodes fondées sur les graphes, l'algorithme PageRank pondère les nœuds (dont ceux du graphe du domaine). La pondération du graphe du domaine permet donc à cette méthode de retourner des mots-clés absents du document, c'est une des rares méthodes en chaîne de traitement à permettre cela.

2.3.3 Sélection du sous-ensemble de mots-clés

Une fois les mots-clés candidats pondérés, il est nécessaire de sélectionner un sous-ensemble de candidats que l'on va considérer comme mots-clés. L'approche la plus simple consiste à sélectionner les n candidats ayant les meilleurs scores. Le choix du n peut être lié entre autre à la longueur du document ou l'utilisation qui sera faite des mots-clés. Ce sont, le plus souvent, les 5 ou 10 premiers mots-clés qui sont choisis, ces chiffres correspondent au nombre moyen de mots-clés annotés par les auteurs et les indexeurs professionnels (cf. section 4.1). La problématique du choix du nombre de mots-clés à associer à un document n'est pas résolue. Certaines méthodes proposent de choisir ce nombre en fonction de la longueur du document (Mihalcea and Tarau, 2004), ou d'entraîner un modèle à produire le bon nombre de mot-clés (Yuan et al., 2018; Chen et al., 2020, *inter alia*).

Lors du choix des mots-clés une étape de filtrage peut être exécutée pour supprimer les mots-clés redondants (Hasan and Ng, 2014). Les mots-clés qui sont contenu dans un mot-clé mieux classé sont généralement redondants. Par exemple si le mot-clé « indexation libre » est mieux classé que « indexation », ce dernier sera considéré comme redondant.

10. Construit de la même manière que pour TopicRank.

2.4 Conclusion

Nous avons présenté dans ce chapitre les concepts importants posant le contexte de cette thèse ainsi que les méthodes de production automatique de mots-clés en chaîne de traitement.

L'annotation de mots-clés, pour être effectuée manuellement, requiert de nombreuses ressources en terme d'indexeurs professionnels, de temps et de moyens financiers. C'est pourquoi la production automatique de ces mots-clés est un enjeu important pour les bibliothèques numériques scientifiques.

Les méthodes précurseures de cette tâche sont dites « en chaîne de traitement » car elles consistent en trois étapes : l'identification de mots-clés candidats, leur pondération puis la sélection d'un ensemble de mots-clés. Chacune de ces étapes est cruciale pour la suivante, si les mots-clés candidats sélectionnés sont peu qualitatifs alors les mots-clés en sortie le seront aussi. La communauté scientifique propose de nombreuses méthodes qui exploitent des caractéristiques telles que la fréquence, la position et la centralité des mots pour extraire les mots-clés des documents. Malgré les avancées de ces méthodes, les performances sont limités par la propagation des erreurs, inhérente aux méthodes en chaîne de traitement, ainsi que le choix des traits utilisé, issus de connaissance expertes. C'est pourquoi des méthodes de bout-en-bout, que nous décrivons dans le chapitre 3, ont été développées.

Chapitre 3

Production de mots-clés de bout-en-bout

Ce chapitre présente les méthodes de production de mots-clés de l'état de l'art de bout-en-bout, qui constituent l'élément principal de ce travail de thèse. Nous commencerons par présenter les composants de ces méthodes auxquels nous feront référence dans la partie état de l'art.

3.1 Principes fondamentaux des réseaux de neurones

Nous présentons dans cette section les concepts nécessaires à la bonne compréhension des méthodes de production de mots-clés de bout-en-bout. Nous décrivons d'abord en détail les réseaux de neurones, ensuite les plongements de mots (*word embeddings*) utilisés pour représenter les mots du langage naturel, et enfin le paradigme encodeur-décodeur qui permet de traiter du texte de longueur variable en entrée et en sortie des réseaux de neurones.

Contrairement aux méthodes en chaîne de traitement décrites dans la section 2.3, les méthodes de bout-en-bout, qui utilisent le paradigme encodeur-décodeur, se passent de l'identification des candidats ainsi que du choix des caractéristiques des mots-clés qui serviront à les pondérer. En effet, les méthodes que nous décrivons dans ce chapitre utilisent des réseaux de neurones profonds qui apprennent à extraire automatiquement les descripteurs les plus pertinents.

3.1.1 Réseaux de neurones

Les réseaux de neurones servent à modéliser des fonctions complexes, qui peuvent être non linéaires. D'un point de vue mathématique, il s'agit de modéliser une fonction f qui prend une entrée X et retourne une sortie (une prédiction) $\hat{Y} = f(X)$. Ici, $X \in M_{1,m}(\mathbb{R})$ et $\hat{Y} \in M_{1,n}(\mathbb{R})$ sont des vecteurs de taille m et n respectivement. Ces vecteurs représentent le nombre de paramètres d'entrée de la fonction f.

Un réseau de neurones simple (également appelé perceptron mono-couche) est défini par l'équation 3.1 ci-dessous, dans laquelle W est une matrice de poids $W \in M_{m,n}(\mathbb{R})$ et

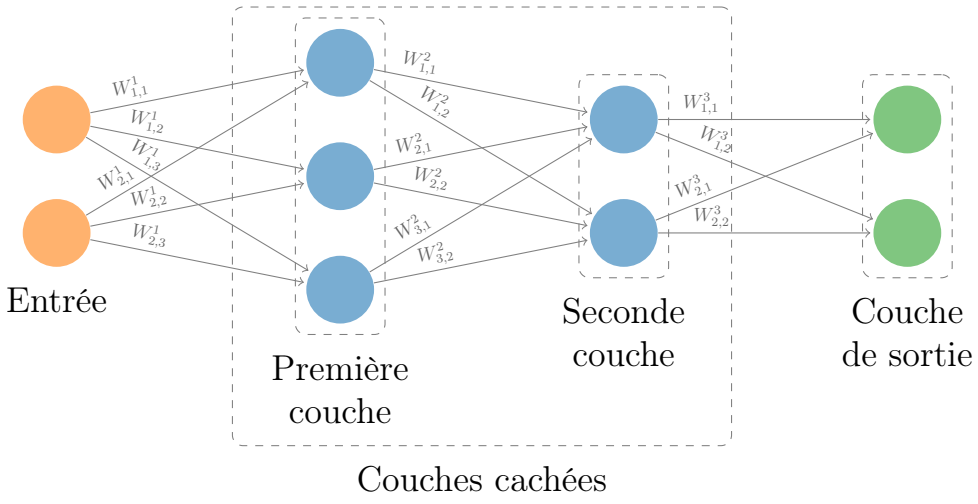


FIGURE 3.1 – Représentation graphique d'un réseau de neurones à 2 couches. Les flèches représentent les poids des matrices W^k. Les biais b^k ne sont pas représentés.

b un vecteur de biais $b \in M_{1,m}(\mathbb{R})$.

$$\hat{Y} = f(X) = \sigma(b + W * X) \tag{3.1}$$

La fonction σ est une fonction dite d'activation. Elle est inspirée de l'activation des neurones dans le cerveau humain. L'information d'un neurone passe à la couche suivante si le neurone est activé (c'est-à-dire si sa valeur est assez élevée). Les fonctions Sigmoïde (voir équation 3.2) et Rectified Linear Unit (ReLU) (voir équation 3.3) sont généralement utilisées pour σ.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

$$\text{ReLu}(x) = \max(0, x) \tag{3.3}$$

Un réseau de neurones à une couche (perceptron mono-couche) ne peut modéliser que des fonctions linéaires, et ne pourra jamais modéliser une fonction telle que la fonction XOR par exemple (Minsky and Papert, 1988). Pour gagner en expressivité, des couches de neurones sont ajoutées (voir figure 3.1). Un tel réseau de neurones est aussi appelé perceptron multicouche. Un perceptron mono ou multicouche dénote un réseau de neurones à propagation avant (*feed forward neural network*) où chaque couche est entièrement connectée à la suivante.

En utilisant les perceptrons multicouche, il est possible de modéliser des fonctions plus complexes. Dans ce cas, chaque couche utilise la sortie de la couche précédente comme entrée (voir équation 3.4). La couche 0 représente l'entrée du réseau, k représente une couche intermédiaire et n la couche de sortie. Les matrices W (et leur vecteurs de biais)

définissent la taille de chaque couche (c'est-à-dire le nombre de neurones). En effet, une matrice de taille $n * m$ (et un vecteur b de taille m) définit une couche de taille m, n étant conditionné par la taille de la couche précédente.

$$\begin{aligned} h^0 &= X \\ h^k &= \sigma(b^k + W^k * h^{k-1}) \\ \hat{Y} &= h^n \end{aligned} \tag{3.4}$$

Entraîner le réseau de neurones f consiste à modifier les paramètres du modèle θ (matrices de poids W^k et b^k) afin de minimiser l'erreur de prédiction. Cette erreur est définie par une fonction de coût (*loss function*) $\mathcal{L}$ (voir équation 3.5) calculé grâce à une prédiction $\hat{Y} = f(X)$ et une vérité terrain Y. Le but de l'entraînement est de minimiser cette fonction de coût pour tous les exemples d'un jeu de données, dénoté par l'équation 3.6.

$$\text{erreur} = \mathcal{L}(\hat{Y}, Y) = \mathcal{L}(f(X, \theta), Y) \tag{3.5}$$

$$\min_\theta \sum_{(x,y)\in\mathcal{D}} \mathcal{L}(f(x, \theta), y) \tag{3.6}$$

L'algorithme utilisé pour minimiser la fonction de coût dans le cadre des réseaux de neurones est la descente de gradient (Curry, 1944) (Gradient Descent). Le gradient de la fonction de coût donne la direction dans laquelle modifier les paramètres (ici les poids du réseau de neurones) pour minimiser ce coût. Cet algorithme calcule le gradient avec tous les exemples du jeu de données puis met à jour les poids du réseau ; ce processus est donc très long. Pour pallier cette lenteur, la descente de gradient par minibatch calcule le gradient à l'aide d'un nombre fixe d'exemples qui fait un compromis entre temps de calcul et utilisation d'un grand nombre d'exemples.

Les réseaux de neurones que nous venons de décrire fonctionnent avec une entrée de taille fixe X ; or dans le cas du texte, les entrées sont des séquences de mots de longueur variable, et n'ont donc pas de longueur fixe. Nous présentons dans la suite de cette section une architecture permettant de traiter les entrées de longueur variable.

3.1.2 Encodage de séquences (encodeur)

Représentation dense des mots

Les réseaux de neurones manipulent des vecteurs et des matrices de nombres. Le langage naturel étant constitué par des mots, il est nécessaire de transformer les mots en nombres. Pour cela, un vocabulaire qui associe un nombre à chacun des mots est nécessaire. La méthode la plus simple est de considérer les mots les plus fréquents d'un ensemble de documents d'entraînement comme le vocabulaire $\mathcal{V}$. Un vocabulaire de taille

trop importante peut être difficile à modéliser et contiendra de nombreux hapax [1]. Il est donc courant de conserver les 50 000 mots les plus fréquents (Meng et al., 2017; Yuan et al., 2020; Ye et al., 2021b).

Aujourd'hui, les plongements de mots sont l'état de l'art en termes de représentation de mots. Ce sont des représentations denses apprises sur de grandes quantités de données qui modélisent les cooccurrences entre les mots : deux mots utilisés dans des contextes similaires auront des vecteurs similaires (Mikolov et al., 2013). Plusieurs méthodes ont été proposées pour pré-entraîner ces plongements. Les méthodes précurseures word2vec (Mikolov et al., 2013) et gloVe (Pennington et al., 2014) calculent des représentations fixes des mots. Leurs principales lacunes sont la prise en compte de mots hors du vocabulaire et la prise en compte de la polysémie. Pour prendre en compte les mots hors du vocabulaire, fastText (Bojanowski et al., 2017) apprend des plongements de n-grammes de caractères puis combine ceux-ci pour représenter le mot entier. Ensuite, pour traiter la polysémie, Elmo (Peters et al., 2018) et Bert (Devlin et al., 2019) calculent ces plongements en fonction du contexte des mots. Il est aussi possible d'entraîner des plongements à l'aide d'une simple matrice de poids au début du réseau de neurones. Dans la tâche de génération de mots-clés, cette technique est la plus utilisée.

Encodeurs

Les encodeurs sont des réseaux de neurones qui permettent de traiter des séquences de longueur variable. Dans nos travaux ces séquences sont des documents textuels. Les encodeurs prennent en entrée une séquence de vecteurs, ici des plongements de mots, et donnent en sortie un vecteur de pensée qui agrège l'information de la séquence d'entrée.

Il existe différents types de réseaux permettant d'encoder une séquence : les réseaux récurrents, les réseaux à convolution, les transformers, les réseaux à convolutions de graphes. Nous nous intéresserons ici aux encodeurs récurrents, les plus utilisés pour la génération de mots-clés. Un encodeur récurrent est composé d'une ou plusieurs cellules récurrentes empilées. Une cellule récurrente RNN encode une séquence selon l'équation 3.7.

$$\begin{aligned} h_t &= \text{RNN}^e(x_t, h_{t-1}) \\ \text{RNN}^e(x_t, h_{t-1}) &= \tanh(W_x * x_t + b_x + W_h * h_{t-1} + b_h) \end{aligned} \tag{3.7}$$

Dans l'équation 3.7 x_t représente le plongement du mot t, h_t l'état caché au temps t qui encode la sous-séquence d'entrée jusqu'au mot t, W_x et W_h des matrices de poids et b_x, b_h les vecteurs de biais correspondants. Le premier état caché h_0 est initialisé aléatoirement. Ces réseaux sont dits récurrents car l'état caché h_t est conditionné par les états cachés précédents.

Ce type d'encodeurs peut être utilisé pour entraîner un classifieur de documents en utilisant le dernier vecteur h_t, par exemple dans le cadre de la catégorisation de docu-

1. Mots n'apparaissant qu'une seule fois.

ments (Wang, 2018) ou la détection de rumeurs (Ma et al., 2016). Il est aussi possible d'utiliser ces encodeurs pour entraîner des modèles d'annotation en séquence qui utilisent chaque représentation h_t pour prédire l'étiquette du mot, par exemple dans le cadre de la reconnaissance d'entités nommées (Žukov Gregorič et al., 2018).

Le principal écueil des réseaux récurrents est leur difficulté à garder en mémoire les dépendances à long terme. Pour pallier ce problème, Hochreiter and Schmidhuber (1997) introduit les cellules LSTM (Long Short Term Memory). Ces cellules récurrentes contiennent des « portes » qui leur permettent d'apprendre à sélectionner les informations à conserver ou à oublier à chaque temps t. Elles donnent en sortie un état caché h_t qui représente l'information pertinente au moment t ainsi qu'un vecteur de contexte C_t qui contient l'ensemble de l'information que la cellule a mémorisé. Ce mécanisme de portes permet de prendre en compte les dépendances à long terme. Cette cellule récurrente a été améliorée, notamment par les cellules récurrentes GRU (Gated Recurrent Unit)(Cho et al., 2014) qui simplifient les LSTM en combinant h_t et C_t en un unique vecteur ainsi qu'en diminuant le nombre de paramètres utilisés, ce qui en facilite l'entraînement.

Pour améliorer l'encodage des documents et la modélisation des dépendances à long terme, Schuster and Paliwal (1997) présente des encodeurs bi-directionnels. Ceux-ci sont composés de deux encodeurs qui encodent chacun la séquence dans un sens différent, du début à la fin et inversement. Chacun de leurs états cachés sont ensuite concaténés pour n'en former qu'un. Ainsi, les états cachés de ce type d'encodeur représentent la séquence entière centrée sur le mot t.

3.1.3 Génération de séquences (décodeur)

Le processus de décodage permet de générer une séquence de mots à partir d'un vecteur de pensée. Ce vecteur de pensée résulte généralement de l'encodage d'un document (cf. section 3.1.2). Comme pour l'encodage, différents types de réseaux peuvent être utilisés : les réseaux récurrents, les réseaux à convolutions ou les transformers. Nous nous intéressons ici aux réseaux récurrents, présentés dans la section 3.1.2.

Un décodeur récurrent est composé d'une ou plusieurs cellules récurrentes empilées. La génération d'une séquence consiste à utiliser les états cachés h_t pour prédire un mot, c'est-à-dire à produire une distribution de probabilités sur l'ensemble du vocabulaire puis à choisir le mot le plus probable.

Le processus de décodage est décrit par l'équation 3.8 avec y_{t-1} le mot prédit précédemment, h_{t-1} l'état caché précédent, W_x et W_h des matrices de poids et leurs biais correspondant b_x et b_h. Le premier mot y_0 est initialisé par un mot spécial ; l'état caché h_0 est un vecteur de pensée. Puis $p(y_t|y_{1,\dots,t-1}, h_0)$ représente la distribution de probabilités sur le vocabulaire pour le mot t en fonction des mots précédents et du vecteur de pensée

h_0 ; W_v et b_v sont une matrice de poids et son biais.

$$\begin{aligned} p(y_t|y_{1,\dots,t-1}, h_0) &= \textsc{Softmax}(\sigma(W_v * h_t + b_v)) \\ h_t &= \textsc{Rnn}^d(y_{t-1}, h_{t-1}) \\ \textsc{Rnn}^d(x_t, h_{t-1}) &= \tanh(W_x * x_t + b_x + W_h * h_{t-1} + b_h) \end{aligned} \tag{3.8}$$

La génération d'une séquence de mots s'effectue mot-à-mot grâce à un algorithme de décodage. Chaque étape de décodage consiste à produire une distribution de probabilité sur le vocabulaire de sortie $p(y_t|y_{1,\dots,t-1}, h_0)$ et à choisir un mot $\hat{y}_t$ qui maximise cette probabilité. Cette étape est répétée jusqu'à ce qu'un mot spécial finissant la séquence soit généré ou que la séquence soit assez longue.

L'algorithme le plus simple consiste à choisir le mot le plus probable à chaque étape mais cela ne permet de générer qu'une seule séquence. Dans le cadre de la génération de mots-clés il est souhaitable de générer plusieurs mots-clés et donc plusieurs séquences. Pour cela, c'est l'algorithme de recherche en faisceau (Ow and Morton, 1988), que nous schématisons dans la figure 3.2, qui est utilisé. Cet algorithme consiste à décoder un nombre fixe de séquences (deux dans le schéma). Les étapes de décodage sont effectuées pour chacun des faisceaux et les mots à générer sont choisis dans l'ensemble des mots des faisceaux. Dans la figure 3.2, deux mots sont choisis au départ. Une étape de décodage est ensuite effectuée pour les deux faisceaux. À la troisième étape, le préfixe `ba` n'est pas conservé car les deux séquences les plus probables sont issues du préfixe `ab`. À la fin de l'algorithme, deux séquences ont été générées : `aba` et `abb`.

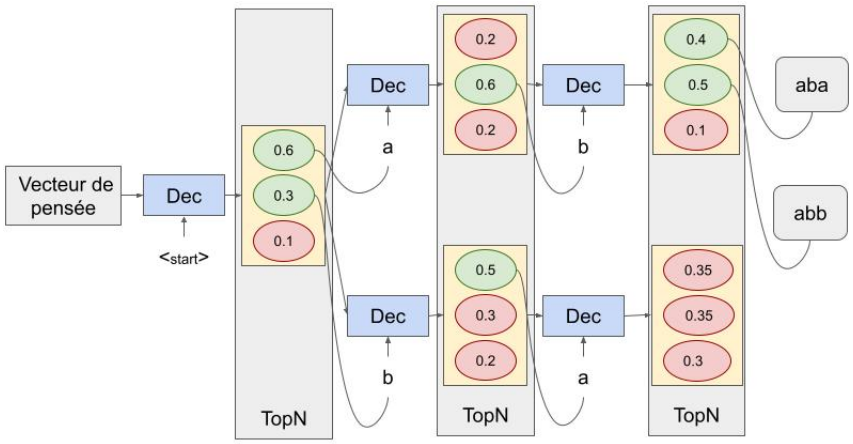


FIGURE 3.2 – Représentation schématique du processus de décodage grâce à l'algorithme de recherche en faisceau.

À l'inverse des algorithmes déterministes que nous avons décrits, notons l'existence des algorithmes d'échantillonnage. Ces algorithmes stochastiques choisissent les mots au hasard en fonction de leur probabilité. Les mots-clés générés par ces algorithmes ne sont donc pas reproductibles, ce qui dans le cadre de la production de mot-clés n'est pas

souhaitable. Nous ne considérons donc pas cette technique.

3.1.4 Paradigme encodeur-décodeur

Nous avons présenté dans les sections 3.1.2 et 3.1.3 des moyens d'encoder des documents ainsi que des moyens de générer des séquences. De nombreuses applications du traitement automatique de la langue nécessitent à la fois l'encodage d'une séquence et son décodage. Par exemple dans le cadre de la traduction automatique, étant donnée une phrase en langue source, il faut la traduire dans une langue cible, c'est-à-dire qu'il faut encoder la phrase dans la langue source puis générer une phrase correspondante dans la langue cible. Dans le cadre de la production de mots-clés, il faut générer un mot-clé en fonction d'un document. Ainsi, le paradigme encodeur-décodeur introduit par Sutskever et al. (2014), qui concatène un encodeur et un décodeur, permet de prendre en entrée une séquence de mots de longueur variable et de générer en sortie une autre séquence de mots de longueur variable. Ce paradigme pallie la limite des réseaux de neurones présentés dans la section 3.1.1 (perceptrons mono ou multicouches) dont l'entrée et la sortie sont de taille fixe.

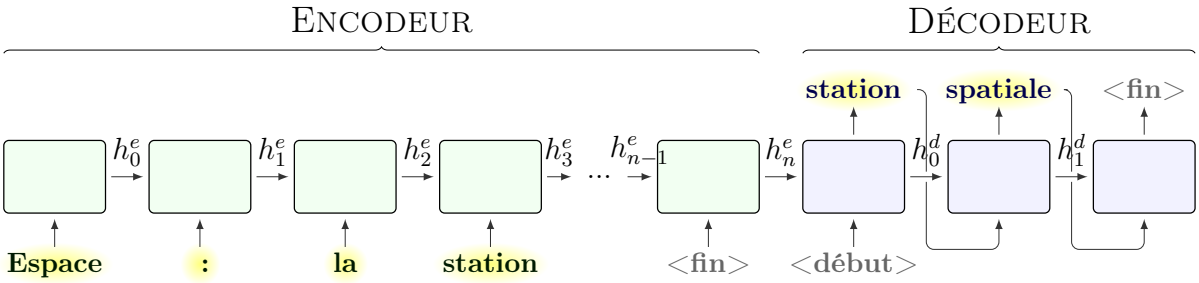


FIGURE 3.3 – Exemple de modèle *encodeur-décodeur* récurrent appliqué à l'extraction automatique de mots-clés.

Le processus d'encodage et de décodage est décrit par l'équation 3.9 et la figure 3.3. Dans un premier temps la séquence d'entrée X de taille n est encodée dans le vecteur de pensée h^e_n. Ce vecteur h^e_n est utilisé pour initialiser le premier état caché du décodeur h^d_0. Le décodeur génère ensuite les mots $\hat{y}_t$ qui composent la séquence de sortie $\hat{Y}$ à partir de cet état caché.

$$
\begin{aligned}
p(\hat{y}_t|y_{1,\dots,t-1}, h_0) &= \textsc{Softmax}(\sigma(b_v + W_v * h^d_t)) \\
h^d_t &= \textsc{Rnn}^d(\hat{y}_{t-1}, h^d_{t-1}) \\
\hat{y}_0 &= \textsc{Debut} \\
h^d_0 &= h^e_n \\
h^e_n &= \textsc{Rnn}^e(X)
\end{aligned}
\tag{3.9}
$$

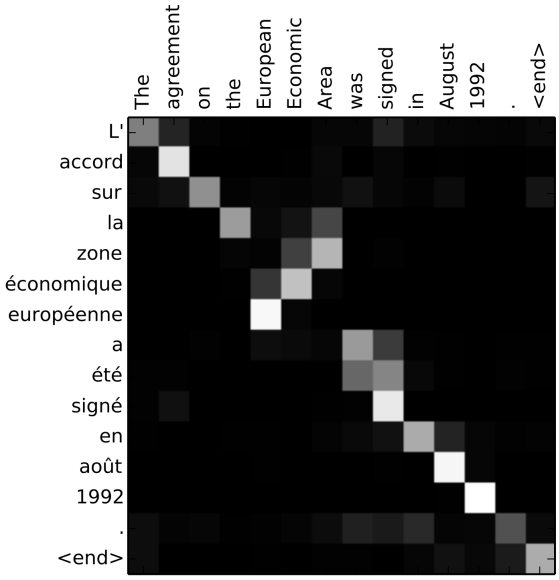


FIGURE 3.4 – Exemple de visualisation des poids d'alignement du mécanisme d'attention entre une phrase en anglais et sa traduction en français. Chaque ligne montre la distribution des poids α_t ayant servis à générer le mot correspondant en français. Une case blanche indique un poids de 1, une case noire indique un poids de 0. Image extraite de Bahdanau et al. (2014).

Nous présentons ci-après deux améliorations de ce paradigme. D'abord, le mécanisme d'attention qui permet de porter attention à une partie spécifique de l'entrée lors du décodage. Par exemple, la description d'une image nécessite d'identifier les différents objets qui la composent. Ensuite, le mécanisme de copie qui pallie l'incomplétude du vocabulaire de sortie. Ce mécanisme permet au décodeur de copier un mot du document d'entrée au lieu de le générer à partir du vocabulaire de sortie. Le mécanisme de copie est particulièrement utile pour les entités nommées par exemple. Ces entités sont peu fréquentes et ne font généralement pas partie du vocabulaire de sortie.

Mécanisme d'attention

Le mécanisme d'attention (Bahdanau et al., 2014; Luong et al., 2015) a été introduit pour améliorer le traitement de longues séquences en permettant au modèle de se focaliser sur certaines parties du document lors du décodage. En effet, un mot-clé concerne seulement certains aspects d'un document. Ce mécanisme permet donc au modèle de porter attention aux parties du document liées à ces aspects. De plus, cette attention au document peut être visualisée grâce aux *poids d'attention* calculés à chaque étape de décodage. La figure 3.4 illustre cette attention dans le cadre de la traduction automatique pour traduire en français la phrase « The agreement on the European Economic Area was signed in August 1992. »

Le décodeur utilise l'état caché courant h_t^d pour générer un mot, le mécanisme d'attention lui permet d'utiliser aussi tous les états cachés de l'encodeur h^e pour mettre à

jour l'état caché du décodeur h_t^d. Ce mécanisme est décrit dans l'équation 3.10. Dans le mécanisme d'attention, les états cachés h^e sont pondérés en fonction de leur importance pour générer le mot $\hat{y}_t$. Cette importance est établie grâce à une fonction d'alignement a qui calcule une similarité entre l'état caché courant h_t^d et ceux de l'encodeur h^e. Les états cachés h^e sont ainsi moyennés dans le vecteur de contexte c_t utilisé pour mettre à jour l'état caché courant du décodeur h_t^d.

Dans l'équation 3.10 : $[u; v]$ représente l'opération de concaténation des vecteurs u et v ; a est une fonction d'alignement qui calcule la similarité entre un état caché de l'encodeur h_t^e et du décodeur h_t^d ; α représente les poids d'alignement entre les états cachés de l'encodeur h^e et du décodeur h^d ; et SOFTMAX est une fonction qui normalise les valeurs d'un vecteur pour qu'il somme à 1.

$$\begin{aligned} p(y_t|y_{<t}, x) &= \text{SOFTMAX}(\sigma(W_v h_t^d)) \\ h_t^d &= \text{RNN}(y_{t-1}, [h_{t-1}^d; c_t] \\ c_t &= \sum_{i=0}^{|h^e|} \alpha_{i,t} h_i^e \\ \alpha_t &= \text{SOFTMAX}(a(h_t^d, h^e)) \end{aligned} \tag{3.10}$$

Mécanisme de copie

Le mécanisme de copie (See et al., 2017; Gu et al., 2016) provient des tâches de traduction automatique et de résumé automatique. Il a pour but de produire des mots peu fréquents ou hors du vocabulaire de sortie. En effet, les modèles neuronaux qui génèrent du texte choisissent les mots dans un vocabulaire de sortie comportant généralement 50 000 mots. Dans les tâches sus-citées, les mots peu fréquents qui ne font pas partie du vocabulaire de sortie, comme les entités nommées ou les transfuges, doivent pourtant apparaître dans la séquence de sortie. Deux mécanismes de copie ont été proposés par See et al. (2017) et Gu et al. (2016) ; les deux étant similaires, nous présentons ici le premier car plus simple. Il est décrit dans l'équation 3.11.

Ce mécanisme utilise le vocabulaire de la séquence d'entrée $\mathcal{X}$ (particulier à chaque document) en plus du vocabulaire de sortie $\mathcal{V}$. Pour produire un mot, une distribution de probabilité sur le vocabulaire $P_{vocab}(y_t)$ est calculée comme précédemment par le décodeur (cf. section 3.1.3) et les poids du mécanisme d'attention α (cf. section 3.1.4) sont utilisés pour estimer la probabilité de copie de chaque mot du document. Les poids d'attention α des mots qui apparaissent plusieurs fois dans l'entrée x sont sommés $\sum_{j,x_j=y_{t,i}} \alpha_j^t$. Ainsi, un mot peut être généré à partir du vocabulaire de sortie $\mathcal{V}$ ou copié à partir du vocabulaire du document $\mathcal{X}$. Les probabilités de copie et de génération d'un même mot, qui appartient au document et au vocabulaire, sont sommées. Dans l'équation 3.11 : h_t^d, c_t et α_j^t proviennent du mécanisme d'attention (cf. équation 3.10) ; p_{gen} est un curseur permettant au modèle

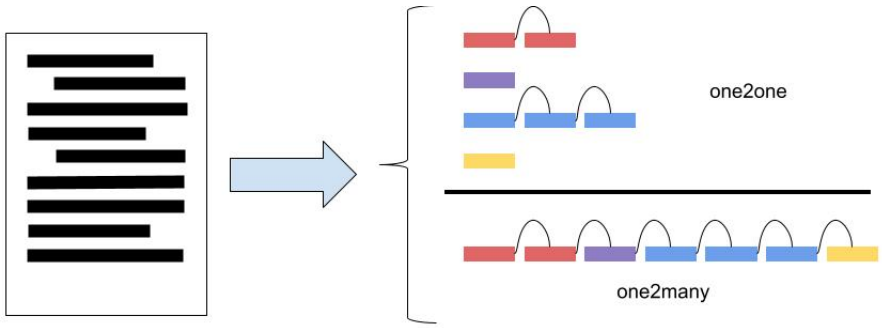


FIGURE 3.5 – Représentation schématique des stratégies de décodage *one2one* et *one2many*.

de privilégier la copie ou la génération et $P_{vocab}(y_t)$ est une distribution de probabilité sur le vocabulaire de sortie $\mathcal{V}$.

$$\begin{aligned} p(y_{t,i}|y_{<t}, x) &= p_{gen} P_{vocab}(y_{t,i}) + (1 - p_{gen}) \sum_{j, x_j = y_{t,i}} \alpha_j^t \\ p_{gen} &= \sigma(W_h h_t^d + W_c c_t + W_y y_{t-1}) \\ P_{vocab}(y_t) &= \text{SOFTMAX}(\sigma(W_v h_t^d)) \end{aligned} \tag{3.11}$$

3.2 Méthodes de bout-en-bout

Dans cette section nous présentons un état de l'art des méthodes de bout-en-bout. Ces méthodes, contrairement aux méthodes en chaîne de traitement (cf. section 2.3), prennent en entrée un document et laissent le soin au modèle d'en extraire les caractéristiques pour retourner un ensemble de mots-clés sans étapes intermédiaires ni définition manuelle de ces caractéristiques. Parmi les méthodes proposées dans la littérature, nous distinguons les méthodes génératives, qui peuvent produire des mots-clés présents et des mots-clés absents, des méthodes extractives, limitées aux mots-clés présents.

Jusqu'à présent, toutes les méthodes de bout-en-bout qui ont été proposées sont supervisées et reposent sur des réseaux de neurones (cf. section 3.1.1) qui nécessitent de grandes quantités de données annotées pour être entraînées. Le développement de ces méthodes démarre avec l'introduction du jeu de données KP20k et de la méthode générative CopyRNN par Meng et al. (2017). Le jeu de données KP20k, qui comporte ≃550 000 documents, comble un manque. En effet, seuls de petits jeux de données (de l'ordre du millier de documents) étaient jusqu'alors disponibles. Ce travail a ainsi lancé une nouvelle direction de recherche sur les méthodes génératives de production de mot-clés.

Dans cet état de l'art, nous présentons tout d'abord les méthodes automatiques de génération de mots-clés de bout-en-bout, qui sont au cœur de ce travail de thèse. Nous présentons ces méthodes de génération en deux parties : premièrement, les méthodes

qui génèrent les mots-clés un à un (*one2one*), et deuxièmement, celles qui génèrent des séquences de mots-clés (*one2many*). Ces deux types de génération sont schématisés dans la figure 3.5. Nous présentons ensuite les méthodes extractives de bout-en-bout, c'est-à-dire celles qui se limitent aux seuls mots-clés présents.

3.2.1 Génération de mots-clés

Les méthodes génératives, introduites par Meng et al. (2017), ont pour objectif de pallier deux faiblesses qui concernent la majorité des méthodes extractives présentées précédemment : l'impossibilité de produire des mots-clés absents ainsi que la faible prise en compte de la sémantique. Le paradigme encodeur-décodeur sur lequel les méthodes génératives sont fondées permet d'encoder la sémantique du document. Ainsi, les mots-clés produits sont le fruit d'une « compréhension » du document, contrairement aux méthodes en chaîne de traitement qui s'intéressent à l'« importance » des mots dans le document indépendamment de leur sens. Ces méthodes génératives rendent possible la production de mots-clés absents grâce à la manière dont le décodeur génère la séquence de sortie. Ce processus s'effectue en choisissant, à chaque étape de décodage, un mot à partir d'un vocabulaire de sortie qui est plus grand et différent du vocabulaire du document. Ces méthodes génératives apprennent à générer des mots-clés un par un (génération *one2one*, voir figure 3.5), c'est-à-dire que chaque document X et son ensemble de mot-clés Y de taille N forment un couple $(X, \{Y_0, ..., Y_N\})$, décomposé en autant d'exemples d'entraînement que de mots-clés, $(X, Y_0), ..., (X, Y_N)$.

La méthode pionnière de génération automatique de mots-clés appliquée aux documents scientifiques est CopyRNN (Meng et al., 2017). L'architecture neuronale de cette méthode s'inspire du processus d'annotation humain qui consiste à lire le document pour le comprendre dans son entièreté puis à le résumer grâce à des mots-clés. Pour reproduire ce processus, CopyRNN utilise le paradigme encodeur-décodeur, que nous avons présenté dans la section 3.1.4, pour encoder un document et le décoder ensuite en un mot-clé. Pour améliorer les performances des modèles encodeur-décodeur, il est commun d'utiliser un mécanisme d'attention (voir section 3.1.4). Ce mécanisme permet au modèle de porter attention à certaines parties du document lors de la génération d'un mot. Un mécanisme de copie est aussi ajouté au modèle pour lui permettre de générer des mots peu fréquents (voir section 3.1.4). Ce mécanisme de copie modifie le décodage en permettant de générer un mot à partir du vocabulaire de sortie ou bien à partir du document.
Cette méthode obtient des performances bien plus élevées que les précédentes méthodes extractives. Les performances de CopyRNN sont de l'ordre de 30 points de F-mesure pour les mots-clés présents tandis que les performances des méthodes extractives sont généralement en dessous de 20 points de F-mesure. Les mots-clés absents, qui ne pouvaient jusque-là pas être produits, correspondent peu à la référence : parmi les 50 meilleurs mots-clés absents un seul apparaît dans la référence.

Certaines méthodes proposées essaient d'améliorer l'encodage du document. Chen

et al. (2019b), par exemple, constate que les mots-clés ne sont pas uniformément distribués dans les documents. En particulier 60 % des mots-clés de référence ont au moins un mot en commun avec le titre du document. Pour prendre cela en compte, ils proposent TGNet (Title Guided Network), qui étend CopyRNN en introduisant un nouvel encodeur spécifique au titre, en plus de l'encodeur du document. Cet encodage du titre permet de donner un poids supplémentaire à l'information qu'il contient. Ces deux représentations (du titre et du document) sont ensuite combinées puis fournies au décodeur. Cette méthode améliore nettement les performances de génération des mots-clés présents et absents par rapport à CopyRNN (+5 % sur KP20k).

La redondance dans les ensembles de mots-clés produits est un problème récurrent dans les méthodes de production de mots-clés. En effet, Hasan and Ng (2014) montrent que 8 à 12 % des erreurs des méthodes sont liées à la redondance des mots-clés. Ainsi, les méthodes en chaîne de traitement mettent en place des stratégies, notamment lors de la sélection du sous-ensemble de mots-clés, pour limiter cette redondance (voir section 2.3.3). Dans cette ligne de recherche, Zhao and Zhang (2019) remarquent les méthodes de bout-en-bout ne sont pas exemptes de ce problème, ils s'intéressent ainsi au chevauchement entre les mots-clés générés et ceux de référence. Par exemple, 23,98 % des mots-clés unigrammes générés par CopyRNN font partie d'un mot-clé de référence, et 47,15 % des mots-clés 4-grammes générés par CopyRNN contiennent un mot-clé de référence. Dans l'optique de limiter ces chevauchement, ils présentent le modèle ParaNet$_T$+CoAtt qui entraîne le modèle, à générer à la fois les mots-clés et leurs étiquettes morphosyntaxiques, ainsi la syntaxe des mots-clés générés sera similaire à celle des mots-clés de référence. Pour cela ils ajoutent au modèle CopyRNN un encodeur, pour les étiquettes morphosyntaxiques des mots du document, ainsi qu'un décodeur, pour celles du mot-clé. [2] Les informations des deux décodeurs sont ensuite combinées et utilisées pour générer les mots-clés et leurs étiquettes morphosyntaxiques.

Dans l'optique de reproduire l'annotation humaine, Chen et al. (2019a) propose la méthode KG-KE-KR-M qui produit un ensemble de mots-clés en combinant différentes méthodes : génération de mots-clés, extraction de mots-clés, récupération de mots-clés (voir figure 3.6). Dans un premier temps, cette méthode récupère les mots-clés de référence des K documents d'entraînement les plus proches du document traité (grâce à la distance de Jaccard). Ces mots-clés *récupérés* sont concaténés puis encodés. Ils serviront à conditionner la génération de mots-clés. Dans un second temps, des mots-clés sont *extraits* du document en classifiant chaque mot comme mot-clé ou non mot-clé. Ensuite, des mots-clés sont *générés* à partir du document ainsi que des mots-clés récupérés et des mots-clés extraits. Enfin, les mots-clés récupérés, extraits et générés sont pondérés grâce à un classifieur. Cette méthode à la particularité de combiner les méthodes en chaîne de traitement (sélection de candidats puis pondération) et les méthodes de bout-en-bout (apprentissage conjoint de la génération et de l'extraction). Malgré la grande diversité dans les techniques de production de mots-clés candidats, les performances ne sont pas signifi-

2. Les étiquettes morphosyntaxiques du document et des mots-clés proviennent de l'outils Stanford CoreNLP.

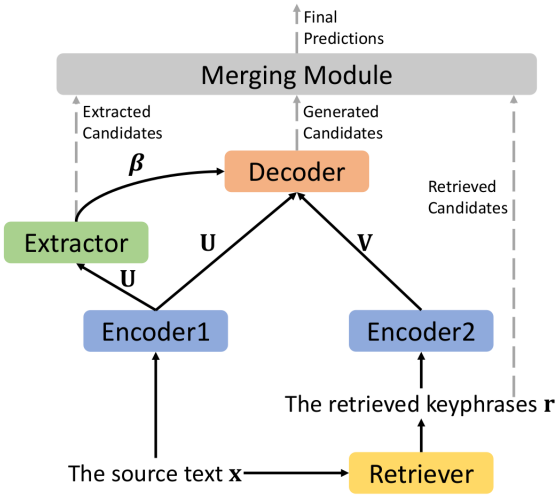


FIGURE 3.6 – Représentation schématique de l'architecture de la méthode KG-KE-KR-M. Image extraite de Chen et al. (2019a).

cativement supérieures à CopyRNN. Cette méthode produit néanmoins plus de mots-clés absents de référence que CopyRNN.

La méthode CorrRNN (Chen et al., 2018) considère que les mots-clés doivent couvrir l'ensemble des sujets du document et être divers, c'est-à-dire que chaque mot-clé doit concerner un sujet différent. Cette méthode étend CopyRNN en y ajoutant un mécanisme de couverture et un mécanisme de revue. Le mécanisme de couverture encourage le modèle à porter attention aux différentes parties du document. Il conserve et accumule les scores d'attention des mots du document à chaque étape de décodage, et il est inclus dans le calcul du mécanisme d'attention. Ensuite, le mécanisme de revue est essentiellement un mécanisme d'attention sur les mots générés. Son objectif est d'identifier les sujets déjà couverts par les mots-clés générés et ainsi de générer des mots-clés qui concernent des sujets non traités. Cette méthode est la première à prendre en compte les mots-clés déjà générés dans le processus de génération, pour cela la phase d'entraînement est modifiée. Au lieu de rétro-propager le gradient après chaque mot-clé de référence, la phase de rétro-propagation n'est effectuée qu'une fois tous les mots-clés de référence du document traités.

3.2.2 Génération de séquences de mots-clés

Nous présentons dans cette section des méthodes qui apprennent à générer des séquences de mots-clés (génération *one2many*, voir figure 3.5). C'est-à-dire que chaque exemple d'entraînement est composé d'un document et de la concaténation des mots-clés de référence en une unique séquence dans laquelle ils sont séparés par un symbole de séparation. Par exemple, l'ensemble de mots-clés { Classe , Fichier log , Agrégat } sera transformé en « Classe `SEP` Fichier log `SEP` Agrégat `FIN` ». Le développement des méthodes génératives *one2many* part du constat que les ensembles de mots-clés produits sont souvent redondants (Hasan and Ng, 2014) et que la génération *one2one* ne permet pas de

pallier ce problème. En effet, les méthodes *one2many* font l'hypothèse qu'avec la génération en séquence, le modèle ayant accès aux mots-clés déjà générés, il ne générera pas de mots-clés redondants. Cette méthode de génération permet au modèle de générer le même nombre de mots-clés que la référence, en effet, il apprend en même temps qu'à générer les mots-clés, à placer les séparateurs de mots-clés et le symbole de fin. Ainsi, ces méthodes peuvent générer des mots-clés selon deux stratégies (Yuan et al., 2020) : l'**inférence exhaustive** qui utilise l'algorithme de recherche en faisceau pour sur-générer des mots-clés et ainsi en obtenir un nombre fixe pour chaque document, c'est la stratégie employée par les méthodes génératives *one2one* ; et l'**inférence auto-régulée** (*self-terminating*) dans laquelle le décodage s'arrête lors de la génération du symbole de fin, cette stratégie permet au modèle de produire un nombre pertinent de mots-clés pour le document. La seconde stratégie de décodage permet donc de s'affranchir du choix arbitraire du nombre de mots-clés n à produire (voir section 2.3.3).

Pour entraîner ces modèles, les mots-clés sont concaténés, mais ce processus n'est pas trivial. En effet, l'ordre dans lequel les mots-clés sont concaténés influence les performances des modèles. L'étude de Meng et al. (2021) compare différentes manières d'ordonner les mots-clés, telles que : *No-Sort* qui laisse l'ordre par défaut ; *Alpha* qui trie par ordre alphabétique ; *Pres-Abs* qui place les mots-clés présents avant les mots-clés absents. L'étude montre que c'est l'ordre *Pres-Abs* qui donne les meilleures performances.

La première méthode à générer des séquences de mots-clés est catSeqD (Yuan et al., 2020, 2018). L'objectif de cette méthode, similaire à CorrRNN, est d'augmenter la diversité des mots-clés générés. Pour cela, le modèle CopyRNN, utilisé comme base, est augmenté d'un mécanisme de couverture sémantique et de régularisation orthogonale pour former le modèle catSeqD. Le mécanisme de *couverture sémantique* repose sur l'hypothèse que l'ensemble de mots-clés de référence et le document encodent la même information. Ainsi, un nouvel encodeur est entraîné à encoder les mots-clés et à produire la même représentation que pour le document. Il encode la séquence au fur et à mesure de sa génération et l'état cachés qui en résulte conditionne la prédiction du mot suivant, cela contraint les mots-clés générés à être proche sémantiquement du document. Ensuite, les auteurs constatent que les mots générés après les séparateurs de mots-clés sont souvent similaires. Le mécanisme de *régularisation orthogonale* pallie ce problème en diversifiant explicitement les représentations des séparateurs, en pénalisant, dans la fonction de coût, ces représentations si elles ne sont pas orthogonales.

Dans le but de mieux modéliser les ensembles de mots-clés, Chen et al. (2020) s'intéressent à la structure hiérarchique des ensembles de mots-clés. En effet, les méthodes de génération de séquences de mots-clés identifient les mots-clés grâce à des marqueurs générés par le modèle. Cette séquentialité ne permet pas de représenter la hiérarchie entre les mots-clés et les mots qui les composent. Ces travaux se rapprochent de Yuan et al. (2018) qui essaient de rompre la séquentialité en modifiant la représentation des séparateurs de mots-clés avec le mécanisme de régularisation orthogonale. Ainsi, ils présentent la méthode ExHirD (Chen et al., 2020) dans laquelle le décodeur de l'architecture de CopyRNN est

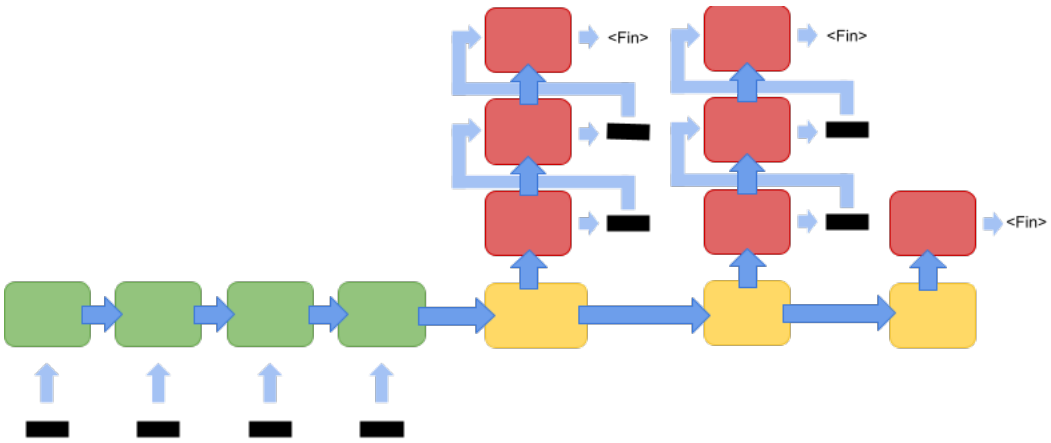


FIGURE 3.7 – Représentation schématique du décodage hiérarchique de la méthode ExHirD. L'encodeur du document est représenté en vert, le décodeur de concept en jaune et le décodeur de mots-clés en rouge.

remplacé par un décodeur hiérarchique (voir figure 3.7) qui génère les mots-clés en deux temps : d'abord l'identification des concepts, ensuite la génération de leur représentation textuelle. Ce décodeur hiérarchique comprend un premier décodeur qui produit une représentation dense d'un concept, puis un second décodeur qui va générer une séquence de mots à partir de cette représentation dense pour instancier le concept en un mot-clé. La génération des mots utilise deux mécanismes d'attention sur les documents d'entrée : l'un est conditionné par la représentation dense du concept ; l'autre, standard, est conditionné par le mot précédent. Ainsi, ce décodeur hiérarchique permet de modéliser explicitement les concepts importants du document et les mots qui les décrivent. L'évaluation de cette méthode montre néanmoins un faible gain de performance, de l'ordre d'un point de F@5, pour les mots-clés présents et absents. Ces travaux s'attellent aussi au problème de redondance des mots-clés et proposent un mécanisme de décodage exclusif pour tenter de le résoudre. Ce mécanisme, simple dans son idée, interdit au modèle de générer deux mots-clés commençant par le même mot. En effet, les mots-clés comportent le plus souvent entre 1 et 4 mots (voir section 2.2.1), ainsi le premier mot affecte grandement les suivants. Ce mécanisme n'est pas limité à la méthode ExHirD ; il peut être adapté aux différents types de décodage ou être utilisé en post-traitement. Son évaluation montre qu'il fait significativement baisser le nombre de mots-clés dupliqués sans faire baisser les scores de F@5.

Les méthodes génératives *one2many* apprennent à déterminer le nombre de mots-clés à produire mais en génèrent trop peu : catSeqD génère en moyenne 4,3 mots-clés par document alors que la référence en est composée de 5,3 en moyenne. Les travaux de Chan et al. (2019) s'intéressent à encourager les modèles à générer plus de mots-clés, en les entraînant à optimiser le rappel et la F-mesure. Or ces métriques ne peuvent être utilisées comme fonction de coût dans l'algorithme de descente de gradient, car elles ne sont pas dérivables. Pour résoudre ce problème, les auteurs proposent d'utiliser l'apprentissage par

renforcement pour affiner [3] des modèles déjà entraînés. Dans l'apprentissage par renforcement (Williams, 1992), un agent produit une série d'actions en suivant une politique (ici la génération de mots grâce à un modèle génératif), puis est récompensé pour chacune des actions. L'algorithme d'apprentissage par renforcement optimise ainsi les poids du modèle (met à jour la politique) en fonction de la récompense. Dans la méthode proposée, la récompense s'adapte selon le nombre de mots-clés générés : s'il est trop faible, la récompense sera le rappel pour encourager le modèle à générer plus de mots-clés ; à l'inverse s'il est trop grand, la récompense sera la F-mesure, pour encourager le modèle à générer seulement de bons mots-clés. De plus, les mots-clés présents et absents sont récompensés séparément pour favoriser la génération des mots-clés absents.

Les travaux, concernant les méthodes neuronales, présentés jusqu'à présent considèrent que la quantité de données disponibles est suffisante. Nous verrons dans le chapitre 4 que les sources de données contenant des documents annotés en mots-clés sont peu nombreuses malgré la large disponibilité de documents scientifiques en ligne. Ainsi, les travaux de Ye and Wang (2018) se placent dans un cadre où la quantité de documents annotés est limitée. Pour cela, les auteurs proposent deux méthodes qui tirent parti de la masse de documents non annotés pour la génération de mots-clés. La première méthode consiste à utiliser des documents non annotés en mots-clés dans le cadre d'apprentissage multitâche. Un réseau de neurones encodeur-décodeur est entraîné, pour les documents annotés, à générer des séquences de mots-clés et, pour les documents non annotés, à générer le titre du document. Dans le modèle, deux décodeurs différents sont utilisés pour chacune des tâches mais l'encodeur est partagé. La seconde méthode consiste à créer un corpus synthétique en annotant automatiquement des documents en mots-clés. Les mots-clés sont extraits grâce aux méthodes Tf×Idf et TextRank. Ainsi, un modèle de génération de mots-clés est pré-entraîné grâce à la combinaison des corpus synthétique et annoté, puis affiné grâce au seul corpus annoté. L'évaluation des deux modèles résultant de ces méthodes d'entraînement montre qu'ils obtiennent des résultats similaires. Les scores de F@5 pour les mots-clés présents des modèles semi-supervisés sont comparables à ceux du modèle *catSeq* (CopyRNN entraîné à générer des séquences de mots-clés), bien qu'ils n'utilisent qu'un dixième des documents annotés utilisés par *catSeq*.

3.2.3 Extraction de mots-clés

Les méthodes génératives de bout-en-bout sont très performantes pour produire des mots-clés présents, mais génèrent très peu de mots-clés absents. Ainsi, la communauté scientifique s'intéresse à des méthodes de bout-en-bout exclusivement extractives. Bien qu'elles ne soient pas au cœur de nos travaux, nous présentons les principales méthodes extractives par soucis d'exhaustivité. Dans cette section nous présentons tout d'abord les méthodes fondées sur l'annotation en séquence, ensuite, une méthode de classification, et enfin, une méthode fondée sur les graphes.

3. *To fine-tune* en anglais.

Le développement de ces méthodes est lié à celui des modèles de langues pré-entraînés tels que BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019) ou encore GPT-2 (Radford et al., 2019) qui reposent sur l'architecture transformer (Vaswani et al., 2017). Ils sont utilisés pour fournir des plongements de mots contextuels ou bien pour être affinés pour une tâche particulière. Ces modèles, entraînés sur de très grandes quantités de données, ont permis d'améliorer significativement les performances de nombreuses tâches de traitement automatique de la langue (Wang et al., 2018).

Annotation en séquence La grande majorité des méthodes extractives de bout-en-bout reformulent la tâche de production de mots-clés en une tâche d'annotation en séquence. Dans l'annotation en séquence, chaque mot du document est associé à une étiquette selon un schéma binaire : mot-clé ou non mot-clé, ou bien selon le schéma `BIO` dans lequel les mots du document correspondent au début (`B`), à l'intérieur (`I`) ou à l'extérieur (`O`) d'un mot-clé.
La méthode pionnière, proposée par Augenstein and Søgaard (2017), utilise un encodeur récurrent bi-directionnel pour représenter chacun des mots et prédire leurs étiquettes. Elle est amélioré par Alzaidy et al. (2019) qui ajoute un champ aléatoire conditionnel (CRF) pour améliorer la prédiction séquentielle des étiquettes, ainsi que par Sahrawat et al. (2019) qui utilise les plongements contextuels de BERT en entrée de l'encodeur. La méthode SaSaKe (Santosh et al., 2020), quant à elle, utilise les relations de dépendances syntaxique et sémantique du document pour améliorer la représentation des mots. Le document est encodé puis les relations de dépendances sont représentées sous formes de graphes et incorporées aux représentations des mots grâce à des réseaux à convolution de graphes. Ces représentation servent ensuite à étiqueter chaque mot comme mot-clé ou non mot-clé.

Classification La méthode BERT-JointKPE (Sun et al., 2020) s'inspire des méthodes en chaîne de traitement pour entraîner un modèle de bout-en-bout à classifier chaque n-gramme du document comme mot-clé ou non mot-clé. Cette méthode ressemble donc à une sélection de mots-clés candidats n-grammes (voir section 2.3.1). Les plongements des mots du document sont d'abord calculés à l'aide de BERT. Ensuite, grâce à des convolutions de différentes tailles, les représentations des mots sont agrégées pour représenter les n-gramme (de 1 à 5). Enfin, chaque n-gramme est classifié comme mot-clé ou non mot-clé grâce à sa représentation dense.

Graphe La méthode DivGraphPointer (Sun et al., 2019) diffère des autres méthodes extractives car elle est fondée sur le paradigme encodeur-décodeur. Nous la décrivons en détail pour comparer son architecture à celles des méthodes génératives décrites dans les sections 3.2.1 et 3.2.2. Cette méthode combine la représentation sous forme de graphe, largement utilisée par les méthodes en chaîne de traitement (voir section 2.3.2), et la

génération de mots-clés en séquence (*one2many*).[4] L'intérêt de cette représentation est double : elle permet premièrement de mutualiser l'information des multiples occurrences d'un même mot ; et deuxièmement, elle permet de prendre en compte les interactions entre les mots de manière globale. Ainsi, le document est d'abord représenté sous forme de graphe dans lequel les nœuds représentent les mots et les arêtes la distance entre les positions des mots. Ensuite, des couches de convolution de graphe calculent la représentation de chaque nœud en fonction de ses voisins. Ces représentations sont agrégées pour initialiser le décodeur, un *pointer network* (Vinyals et al., 2016). Enfin, ce décodeur produit une séquence de mot exclusivement copiée du document. DivGraphPointer à pour objectif, comme *catSeqD*, de produire des mots-clés peu redondants. Ainsi, en plus du mécanisme d'attention et de couverture, le mécanisme de *modification du contexte* (similaire dans son objectif à la *régularisation orthogonale* de *catSeqD*) recalcule l'état caché après avoir généré un séparateur de mot-clé. Cet état caché est calculé en fonction de la représentation du document et de l'ensemble des mots-clés précédemment générés.
Un intérêt peu discuté de cette méthode est sa capacité à produire des mots-clés qui ne sont pas des sous-séquences du document mais dont tous les mots y apparaissent. Ainsi, la dichotomie entre mots-clés présents et mots-clés absents ne semble ne pas convenir à ce type de mots-clés. Nous discuterons la définition de mots-clés présents et de mots-clés absents dans le chapitre 7.

3.3 Conclusion

Dans ce chapitre, nous avons présenté les principes fondamentaux des réseaux de neurones ainsi que le paradigme encodeur-décodeur qui permet d'encoder un document de longueur variable et de générer une séquence de mots. Nous avons ensuite présenté un état de l'art des méthodes de production de mots-clés de bout-en-bout, toutes neuronales, qui reposent à minima sur les encodeurs ou les décodeurs. Pour cet état de l'art, nous avons séparé ces méthodes en deux catégories : les méthodes génératives et les méthodes extractives.

La mise à disposition, par Meng et al. (2017), d'une grande quantité de données annotées permet le développement de méthodes de bout-en-bout pour la production de mots-clés. Ces méthodes de bout-en-bout pallient certains écueils des méthodes en chaîne de traitement, présentées au chapitre 2, notamment la propagation des erreurs entre les différentes étapes et la définition manuelle des traits pour identifier l'importance des mots-clés. Néanmoins, les méthodes de bout-en-bout ne sont pas exemptes de limites : elles nécessitent de grandes quantités de données pour être entraînées ainsi qu'une grande puissance de calcul pour être utilisées.

Les méthodes extractives de bout-en-bout s'inspirent, pour la majorité, de l'annotation

4. Cette méthode est générative, mais ne peut produire de mots-clés absents. En dehors de sa description nous réservons le terme « méthodes génératives » aux seules méthodes pouvant produire des mots-clés absents.

en séquence et entraînent des réseaux de neurones à identifier le début et la fin des mots-clés dans les documents. Ces méthodes sont, de manière générale, plus performantes que les méthodes génératives, ainsi, la spécialisation des méthodes dans l'extraction de mots-clés semble faciliter la tâche.

Les méthodes génératives, qui constituent le cœur de nos travaux, entraînent un réseau de neurones à générer les mots-clés de référence. Elles ont la capacité de produire des mots-clés absents, ce que les méthodes proposées jusqu'alors ne permettaient pas. Ces méthodes ont deux principales faiblesses : elles produisent très peu de mots-clés absents (1,7 en moyenne (Chan et al., 2019)) et produisent des mots-clés très redondants (entre 20 % et 30 % (Chen et al., 2020)). Ainsi, les différentes méthodes présentées ont pour objectif de pallier au moins une de ces faiblesses en ajoutant des mécanismes de diversification des mots-clés, en essayant d'améliorer la modélisation des documents ou en modifiant le processus de décodage. De manière globale, les performances de la tâche de production automatique de mots-clés augmentent peu. Notons tout de même l'amélioration des performances pour les mots-clés *présents* de 33 à 40 points de F@5 sur KP20k entre les premiers travaux de Meng et al. (2017) et ceux, plus récents, de Ye et al. (2021a). Mais, malgré cette augmentation de performance pour les mots-clés présents, les performances pour les mots-clés *absents* ne dépassent pas 5 points de F@5 (Chan et al., 2019; Ye et al., 2021a). Nous verrons dans le chapitre 7 que ces mots-clés absents sont un enjeu important pour la tâche de recherche d'information.

Chapitre 4

Cadre expérimental

Dans ce chapitre, nous nous intéressons d'abord aux différents jeux de données utilisés pour évaluer ou entraîner des méthodes de production automatique de mots-clés ainsi qu'à leurs particularités. Nous présentons ensuite les processus d'évaluation de ces méthodes ainsi que les différentes métriques utilisées. Nous terminons ce chapitre par une discussion sur la représentativité des jeux de données que nous avons utilisés dans nos travaux.

4.1 Jeux de données

Nous présentons dans cette section les jeux de données utilisés dans les travaux traitant de production automatique de mots-clés. Nous avons limité cette étude aux jeux de données utilisés dans ce manuscrit.

Nos jeux rassemblent des documents scientifiques et des documents journalistiques. Bien que ce travail se concentre sur la littérature scientifique, nous exploitons aussi des jeux de données journalistiques, relevant du domaine général, qui permettent de tester les capacités d'adaptation des méthodes à d'autres types de données. Nous distinguons articles et notices scientifiques car bien que les deux soient des documents scientifiques, l'accès aux premiers est souvent régi par un péage (*paywall*) mis en place par les éditeurs, alors que les seconds sont librement accessibles.

Les jeux de données présentés ici sont en grande majorité des documents en langue anglaise, à l'exception de TermITH-Eval qui est en français. Les documents scientifiques sont majoritairement annotés en mots-clés par leurs auteurs contrairement aux articles journalistiques généralement annotés par des lecteurs. Pour garantir la qualité des mots-clés lecteurs, différents processus d'annotation sont mis en œuvre, tels que des guides d'annotation ou des séances d'adjudication itérative qui permettent d'obtenir le consensus sur le choix des mots-clés.

Ces jeux de données sont exploités pour entraîner et évaluer des méthodes de production automatique de mots-clés. Ceux qui servent à évaluer les méthodes d'extraction contiennent peu de documents (un millier en moyenne). Au contraire, les jeux de données introduits pour générer des mots-clés constituent de très larges collections de plusieurs centaines de milliers de documents, par exemple KP20k et KPTimes contiennent respectivement 570 000 et 300 000 documents. Les tableaux 4.1, 4.2 et 4.3 présentent les statistiques des jeux de données décrits dans cette section. Ces statistiques sont calculées

sur les ensembles de test. Le ratio de mots-clés absents est calculé en comparant les formes racinisées (Porter, 1980) des mots du document et des mots-clés, dans le but de prendre en compte les variantes flexionnelles.

Les jeux de données sont présentés par type de documents : notices scientifiques (titre et résumé seulement), documents scientifiques (articles, rapports techniques) et articles journalistiques.

4.1.1 Jeux de données composés de notices scientifiques

Le contenu textuel d'une notice scientifique se résume à son titre et à son résumé. Elle peut comporter aussi des informations bibliographiques apparaissant sous forme de métadonnées. Ce sont ces notices qui sont utilisées pour indexer les documents (Huang et al., 2019). Les notices adoptent un format standardisé qui facilite leur traitement ; elles ne contiennent ni tableaux, ni figures, ni références bibliographiques, etc. Les statistiques des jeux de données décrits dans cette section sont détaillées dans le tableau 4.1. Nous présentons les jeux de données par ordre de publication.

Corpus	Lang.	Ann.	#Entr.	#Test	#mots	#mc	%abs
Inspec (Hulth, 2003)	en	*I*	1 000	500	135	9,8	22,4
KDD (Caragea et al., 2014)	en	*A*	-	755	191	4,1	49,3
WWW (Caragea et al., 2014)	en	*A*	-	1 330	164	4,8	52,0
TermITH-Eval (Bougouin et al., 2016)	fr	*I*	-	400	165	11,8	59,8
KP20k (Meng et al., 2017)	en	*A*	530 K	20 K	176	5,3	42,6
				Avg.	166	7,2	45,2

TABLEAU 4.1 – Statistiques des jeux de données de notices scientifiques. Les mots-clés de référence sont annotés par les auteurs (*A*) ou des indexeurs professionnels (*I*). La table présente le nombre de documents dans les corpus d'entraînement (#Entr.) et de test (#Test) ainsi que le nombre moyen de mots-clés (#mc), de mots (#mots) et le ratio de mots-clés absent (%abs) par document.

Inspec (Hulth, 2003) Inspec [1] est une base de données bibliographiques caractérisée par une indexation manuelle réalisée par des indexeurs professionnels. Ce jeu de données contient un ensemble de test, de validation et d'entraînement contenant respectivement 500, 500 et 1 000 documents. Les documents sont en anglais et traitent des domaines « Computers and Control » et « Information Technology ». Deux types d'indexation sont effectués pour ajouter des mots-clés : une indexation contrôlée à l'aide d'un thésaurus, qui garantit une indexation cohérente, et une indexation non contrôlée, c'est-à-dire non restreinte par un ensemble de mots, qui augmente la couverture de l'indexation. Le nombre de mots-clés moyen proposé par l'indexation contrôlée et l'indexation non contrôlée sont respectivement de 4,5 et 9,4.

1. https://www.theiet.org/publishing/inspec

KDD/WWW (Caragea et al., 2014) Ces deux jeux de données sont constitués de, respectivement, 755 et 1 330 notices scientifiques en anglais d'articles provenant des conférences KDD (*Knowledge Discovery and Data mining*) et WWW (*World Wide Web Conference*). Les notices sont associées à des mots-clés auteurs, 4,1 et 4,8 en moyenne par document pour KDD et WWW respectivement. La particularité de ces deux jeux de données est de contenir les contextes de citation des documents qui pourront être pris en compte par les méthodes de production automatique de mots-clés. Ces contextes de citation ont été extraits à partir de la bibliothèque numérique scientifique CiteSeerX [2].

TermiTH-Eval (Bougouin et al., 2016) Dans le cadre du projet ANR TermITH [3], et à l'aide des indexeurs professionnels de l'Inist [4], 400 articles scientifiques en français ont été annotés en mots-clés de manière non contrôlée. Il y a en moyenne 11,8 mots-clés par document. Ces articles scientifiques proviennent de 4 domaines : linguistique, science de l'information, archéologie et chimie. Pour procéder à leur annotation et garantir sa qualité, des principes ont été définis : conformité (utiliser la terminologie du domaine), exhaustivité (identifier tous les mots-clés utiles à la recherche d'information), consistance (deux concepts similaires doivent être représentés par le même mot-clé), spécificité (les mots-clés doivent être le plus précis possible mais des mots-clés plus génériques peuvent être ajoutés) et impartialité (les mots-clés choisis ne doivent pas refléter l'opinion de l'annotateur).

KP20k (Meng et al., 2017) La construction de ce jeu de données est peu documentée. Il contient des notices scientifiques et mots-clés auteurs en anglais qui proviennent de plusieurs bibliothèques numériques scientifiques dont l'ACM Digital Library, ScienceDirect, Wiley, Web of Science [5]. Selon une annotation manuelle de 100 documents pris au hasard dans l'ensemble de test, les documents de KP20k concernent en très grande majorité le domaine de l'informatique (76 %), et dans une moindre mesure les mathématiques (8 %), l'ingénierie (6 %), le domaine médical (5 %), la physique (4 %) et la chimie (1 %). KP20k contient 567 830 documents dont 20 000 sont utilisés comme documents de test ; chaque document comporte en moyenne 5,3 mots-clés. Ce jeu de données a été présenté avec la première méthode supervisée de génération automatique de mots-clés, CopyRNN, qui utilise une architecture neuronale séquence à séquence. C'est le premier jeu de données à contenir suffisamment de documents pour entraîner des méthodes de ce type.

4.1.2 Jeux de données composés d'articles scientifiques

Les jeux de données composés d'articles scientifiques contiennent l'intégralité des documents, c'est-à-dire le titre, le résumé, le corps du texte et la bibliographie. Les statistiques

2. https://citeseerx.ist.psu.edu
3. https://anr.fr/Projet-ANR-12-CORD-0029
4. https://www.inist.fr/
5. dl.acm.org, sciencedirect.com, onlinelibrary.wiley.com, webofknowledge.com

des jeux de données décrits dans cette section sont détaillées dans le tableau 4.2.

Les articles scientifiques sont généralement disponibles au format PDF, ils doivent être convertis au format texte pour être traités par les méthodes de production automatique de mots-clés. L'extraction du texte d'un fichier PDF peut se faire grâce à des techniques de reconnaissance optique de caractères (OCR), ou s'il contient du texte sélectionnable, en reconstruisant le document grâce à la position de ces morceaux de texte. Les articles pleins sont des documents beaucoup plus longs que les seules notices scientifiques : 8 495 mots en moyenne contre 166 pour les notices. Ils sont plus difficiles à traiter de par leur longueur et leur structure (articles double-colonnes, tableaux, sections, etc.). Nous présentons les jeux de données par ordre chronologique.

Corpus	**Lang.**	**Ann.**	**#Entr.**	**#Test**	**#mots**	**#mc**	**%abs**
CSTR (Witten et al., 1999)	en	A	130	500	11 501	5,4	18,7
NUS (Nguyen and Kan, 2007)	en	$A \cup L$	-	211	8 398	10,8	14,4
PubMed (Schutz, 2008)	en	A	-	1 320	5 323	5,4	16,9
ACM (Krapivin et al., 2009)	en	A	-	2 304	9 198	5,3	16,3
Citeulike-180 (Medelyan et al., 2009)	en	L	-	182	8 590	5,4	10,9
SemEval-2010 (Kim et al., 2010)	en	$A \cup L$	144	100	7 961	14,7	19,7
				Avg.	8 495	7,8	16,2

TABLEAU 4.2 – Statistiques des jeux de données d'articles scientifiques. Les mots-clés de référence sont annotés par des auteurs (A) ou des lecteurs (L). La table présente le nombre de documents dans les corpus d'entraînement (#Entr.) et de test (#Test) ainsi que le nombre moyen de mots-clés (#mc), de mots (#mots) et le ratio de mots-clés absent (%abs) par document.

CSTR (Witten et al., 1999) Les documents du jeu de données CSTR proviennent de la New Zealand Digital Library. Ce sont des rapports techniques en anglais dans le domaine de l'informatique. Ces documents sont similaires aux articles scientifiques de par les sujets qu'ils traitent et de par leur structure, mais sont plus longs : 11 501 mots contre 8 495 en moyenne pour les articles scientifiques. CSTR est un des premiers jeux de données ayant servi à évaluer une méthode de production automatique de mots-clés. Il est composé de 630 rapports annotés en mots-clés auteurs avec 5,4 mots-clés par rapport en moyenne. Il est séparé en un ensemble d'entraînement (130 documents) et de test (500 documents), ce qui permet d'entraîner des modèles supervisés.

NUS (Nguyen and Kan, 2007) Le jeu de données NUS est composé de 211 articles scientifiques en anglais qui ont été aspirés du web grâce à la requête `keywords general terms filetype:pdf` qui retourne des documents PDF disponibles dans l'ACM DL. [6] Ces articles scientifiques appartiennent au domaine de l'informatique et sont annotés en mots-clés auteurs. Pour rendre l'évaluation plus robuste, une annotation concurrente effectuée

6. `keywords general terms` sont des mots qui apparaissent dans le style LaTeX de conférences dont les actes sont publiés dans l'ACM DL.

par des étudiants bénévoles a été ajoutée. L'union de ces deux annotations (10,8 mots-clés en moyenne) est utilisée pour évaluer les méthodes de production automatique de mots-clés. Les PDF des articles ont été convertis au format texte à l'aide du logiciel `PDF995` [7].

PubMed (Schutz, 2008) Le jeu de données PubMed contient 1 320 documents en anglais contenant des mots-clés auteurs (5,4 en moyenne) extraits de la bibliothèque numérique scientifique PubMed Central [8]. Cette bibliothèque archive les articles scientifiques publiés dans des revues du domaine biomédical et des sciences de la vie. En plus du format PDF, les articles sont disponibles au format XML, ils sont donc déjà dans un format textuel. Le format XML permet aussi de filtrer les figures et tableaux, et d'utiliser la structure du document.

ACM (Krapivin et al., 2009) Le jeu de données ACM est composé de 2 304 articles scientifiques en anglais provenant du domaine de l'informatique. Chaque document est associé à 5,3 mots-clés auteurs en moyenne. La particularité de ce jeu de données est que les différentes parties des documents ont été identifiées grâce à un traitement automatique. Ces différentes parties sont le titre, le résumé, le corps du texte et la bibliographie. De plus, le texte a été nettoyé de ses formules mathématiques, tableaux et figures à l'aide d'un classifieur.

Citeulike-180 (Medelyan et al., 2009) Ce jeu de données est composé de 180 articles scientifiques en anglais avec en moyennes 5,4 mots-clés annotés par des lecteurs. Le processus d'annotation s'est déroulé en deux étapes au sein de la plateforme CiteULike dédiée à la gestion de bibliographie [9] : (1) les lecteurs assignent les étiquettes de leur choix aux articles ; (2) ne sont conservées comme mots-clés que les étiquettes qui sont assignées au moins deux fois et les documents qui comportent au moins 3 étiquettes. Cette annotation s'inscrit dans une démarche d'annotation non professionnelle similaire au jeu de données NUS (cf. 4.1.2).

SemEval-2010 (Kim et al., 2010) Le jeu de données SemEval-2010 a été constitué à l'occasion de la compétition du même nom. Il contient 244 documents en anglais annotés en mots-clés auteurs extraits de l'ACM DL. Ces documents sont répartis dans quatre domaines de recherches [10] : systèmes distribués ; recherche d'information ; intelligence artificielle distribuée – systèmes multi-agent ; sciences sociales et du comportement – économie. Les documents ont été convertis en texte à l'aide de l'outil `pdftotext` [11]. Dans une démarche similaire à NUS (cf. 4.1.2), les mots-clés auteurs sont complétés par

7. www.pdf995.com
8. www.ncbi.nlm.nih.gov/pmc
9. La plateforme n'est plus accessible depuis 2019.
10. *Distributed Systems*, *Information Search and Retrieval*, *Distributed Artificial Intelligence – Multiagent Systems*, *Social and Behavioral Sciences – Economics*.
11. https://www.xpdfreader.com/pdftotext-man.html

des mots-clés lecteurs qui intègrent des variantes morphosyntaxiques comme *distribution of relevance* et *relevance distribution*. L'annotation a été réalisée par cinquante étudiants recrutés et rémunérés. Pour garantir la qualité de l'annotation, un guide d'annotation a été construit et validé par adjudication. L'union de ces indexations représente en moyenne 14,7 mots-clés par document.

4.1.3 Jeux de données composés d'articles journalistiques

Les articles journalistiques sont pour la plupart extraits de journaux quotidiens en ligne. La taille de ces articles est variable, les brèves sont des documents assez courts ($\simeq$ 100 mots) et les enquêtes peuvent être bien plus longues ($>$ 10 000 mots). Contrairement aux articles scientifiques, les auteurs ne fournissent pas de mots-clés mais il est courant que les articles journalistiques soient classés en catégories générales (sport, politique, culture, ...). Les statistiques des jeux de données décrits dans cette section sont détaillées dans le tableau 4.3. Nous présentons les jeux de données par ordre chronologique.

Corpus	Lang.	Ann.	#Entr.	#Test	#mots	#mc	%abs
DUC-2001 (Wan and Xiao, 2008b)	en	*L*	-	308	847	8,1	3,7
500N-KPCrowd (Marujo et al., 2012)	en	*L*	450	50	465	46,2	11,2
Wikinews (Bougouin et al., 2013)	fr	*L*	-	100	314	9,7	10,8
KPTimes (Gallina et al., 2019)	en	*E*	260 K	20 K	784	5,2	41,4
				Avg.	603	17,3	16,8

TABLEAU 4.3 – Statistiques des jeux de données d'articles journalistiques. Les mots-clés de référence sont annotés par des lecteurs (*L*) ou des éditeurs (*E*). La table présente le nombre de documents dans les corpus d'entraînement (#Entr.) et de test (#Test) ainsi que le nombre moyen de mots-clés (#mc), de mots (#mots) et le ratio de mots-clés absent (%abs) par document.

DUC-2001 (Wan and Xiao, 2008b) DUC-2001 est à l'origine un ensemble d'articles journalistiques en anglais créé pour la campagne d'évaluation éponyme (Over, 2001) destinée à évaluer la tâche de résumé automatique. Les articles proviennent de journaux américains tels qu'Associated Press, le Wall Street Journal, le Financial Times, etc. Les 308 articles qui composent le jeu de données DUC-2001 comportent en moyenne 8,1 mots-clés lecteurs. Leurs mots-clés ont été annotés manuellement par deux étudiants. Le calcul de l'accord inter-annotateur (kappa de Cohen) a montré un accord substantiel (0,7). Les conflits d'annotation ont été résolus par adjudication.

KPCrowd (Marujo et al., 2012) Ce jeu de données est composé de 500 articles journalistiques en anglais provenant de sources multiples peu documentées. Ces articles comportent en moyenne 46,2 mots-clés lecteurs. L'annotation en mots-clés a été effectuée

par des travailleurs de la plateforme de micro-travail Mechanical Turk [12] d'Amazon qui permet la rémunération des annotateurs : $0,02 par document. Chacun des 500 documents a été annoté par 20 annotateurs différents. À la fin de la phase d'annotation, et pour garantir une certaine qualité dans l'annotation, seuls les mots-clés choisis par 90 % des annotateurs sont conservés.

KPTimes (Gallina et al., 2019) Ce jeu de données est composé de 300 000 articles journalistiques en accès libre en anglais provenant du New York Times et du Japan Times. Ces articles comportent des mots-clés éditeur (5,2 en moyenne) proposés par un système automatique d'indexation contrôlée, puis validés par les éditeurs du journal. Ces derniers peuvent aussi ajouter des mots-clés qui n'ont pas été proposés par le système automatique. Ces mots-clés peuvent être associés à des variantes synonymiques telles que *Police Brutality*, *Police Misconduct* et *Police Shootings*. Ce jeu de données fait partie des ressources que nous avons développées. Le processus de construction de ce jeu de données est détaillé dans le chapitre 5.

4.1.4 Autres jeux de données

Les ressources présentées précédemment sont celles que nous utilisons dans nos expériences. Il existe d'autres ressources que nous n'avons pas retenues, comme certains jeux de données présentés dans Yuan et al. (2020) et Campos et al. (2020) qui n'ont pas ou peu été repris dans d'autres travaux. Certains de ces jeux de données auraient pu être intéressants mais ne correspondaient pas à notre cadre expérimental :

— langues différentes de l'anglais ou du français ;

— genres de documents relevant d'autres types d'intentions communicatives comme les courriels ou regroupant de multiples genres de documents comme les pages web ;

— concepts de mots-clés interprétés différemment, par exemple dans le but d'identifier les tâches et outils mentionnés dans des articles scientifiques.

Nous présentons ci-dessous quelques-uns de ces jeux de données.

Bien que la majorité des jeux de données que nous avons présentés soient en anglais (dont un en français), nous notons l'existence de jeux de données dans d'autres langues. Par exemple, 110-PT-BN-KP (Marujo et al., 2011) contient 110 transcriptions de journaux télévisés en portugais associés à des mots-clés lecteurs ; PerKey (Doostmohammadi et al., 2018) contient 550 000 articles journalistiques en persan dont les mots-clés sont annotés par leurs auteurs.

Nous nous sommes focalisés sur les documents scientifiques et journalistiques mais d'autres genres de documents ont été étudiés par la communauté scientifique. Par exemple, Turney (2000) utilise des courriels et des pages web pour évaluer la tâche de production automatique de mots-clés. Plus récemment, Xiong et al. (2019) présente MS-Marco, un

12. https://www.mturk.com

jeu de données de 150 000 pages web annotées par des employés entraînés à associer des mots-clés à des pages web.

Le jeu de données SemEval-2010 (Augenstein et al., 2017) est annoté en mots-clés mais ces derniers ne représentent pas le contenu du document ; ils décrivent les tâches, outils et ressources mentionnées dans l'article.

OAGKX (Çano and Bojar, 2019), un très large ensemble de notices scientifiques (23 millions) avec mots-clés auteurs, a été construit dans le but de pouvoir augmenter des jeux de données existants, ou d'en fabriquer de nouveaux. Ce jeu de données n'a été utilisé à ce jour, à notre connaissance, que par ses auteurs. Il est constitué de notices scientifiques extraites à partir du Open Academic Graph (union de ArnetMiner et Microsoft Academic Graph) et il est peu documenté en termes de métadonnées, en particulier pour identifier les domaines des documents.

4.1.5 Discussion

De nombreux jeux de données ont été proposés par la communauté scientifique, et couvrent un large panel de genres de documents et de types d'annotation. Ce grand nombre de jeux de données traduit un intérêt pour la tâche de production automatique de mots-clés. Malgré cela, la majorité de ces jeux de données contiennent peu de documents et sont annotés de manière non-professionnelle. La petite taille de ces jeux de données ne comble pas le besoin en données d'entraînement des modèles neuronaux, aujourd'hui incontournables. La faible disponibilité d'annotations professionnelles est aussi à déplorer car elle impacte négativement l'évaluation des méthodes ainsi que leur apprentissage (voir section 6.2). Cette faible disponibilité s'explique par le coût financier de telles annotations ainsi que par la difficulté de mobiliser des indexeurs professionnels.

De plus, du fait de la rareté des documents scientifiques annotés en mots-clés, les jeux de données sont généralement construits en utilisant les mêmes sources de documents. Le choix des documents est donc restreint. Par exemple, les jeux de données KP20k, ACM, KDD et WWW ont été en partie construits grâce à la même source de document : l'ACM DL. Il en résulte que ces jeux de données différents contiennent des documents communs. Autrement dit, ils ont une intersection non nulle. Cette intersection non nulle pose problème dans le cas où des documents sont communs à des ensembles de test et à des ensembles d'entraînement. Dans ce cas, l'évaluation de méthodes entraînées sur de tels ensembles d'entraînement est faussée. Pour mesurer l'ampleur de ce phénomène, nous calculons le nombre de documents communs aux différents jeux de données scientifiques. [13] Ainsi nous constatons que 12 % des documents de test d'Inspec apparaissent dans l'ensemble d'entraînement de KP20k, ce chiffre est de 68 % pour KDD, 63 % pour WWW, 6 % pour KP20k, 1 % pour PubMed, 44 % pour ACM et 84 % pour SemEval-2010. Pour KP20k, ce phénomène est mentionné pour la première fois dans les travaux

13. Nous considérons deux documents égaux si leurs titres sont égaux. Les titres ont été mis en minuscules et les caractères de ponctuation ont été supprimés.

de Chen et al. (2019a) et Chan et al. (2019) qui explicitent leurs manières de résoudre ce problème en supprimant de l'ensemble d'entraînement les documents qui apparaissent dans des ensembles de test.

4.2 Évaluation

L'évaluation d'un ensemble de mots-clés prédits s'effectue classiquement de manière intrinsèque. L'évaluation intrinsèque s'intéresse à la pertinence des mots-clés prédits par rapport à des mots-clés de référence. L'évaluation extrinsèque est un autre type d'évaluation et vise à démontrer l'intérêt des mots-clés dans une tâche applicative. Cette évaluation sera présentée dans le chapitre 7. Nous nous concentrons dans cette section sur l'évaluation intrinsèque.

L'évaluation intrinsèque peut être effectuée de manière manuelle ou automatique. Elle vise à déterminer si l'ensemble de mots-clés prédits associés à un document présente bien les caractéristiques requises : non redondance et couverture (voir section 2.2). Dans la réalité, comme nous l'avons vu précédemment, les mots-clés de référence sont annotés par des annotateurs professionnels : les indexeurs, ou par des non professionnels : les auteurs, les lecteurs ou les éditeurs. Les annotateurs professionnels respectent mieux les caractéristiques des mots-clés que les annotateurs non professionnels.

L'évaluation d'une liste ordonnée de mots-clés associée à un document s'effectue en deux étapes : la validation ou le rejet de chaque mot-clé prédit ; puis le calcul d'un score selon ce jugement. Nous présentons dans cette section la première étape qui consiste à comparer les mots-clés prédits à une liste de mots-clés de référence, c'est l'étape d'appariement. Nous présentons ensuite la seconde étape, le calcul du score, qui s'effectue grâce aux métriques classiques de TALN que sont la précision, le rappel et la F-mesure. Enfin, nous présentons une manière de rendre l'évaluation plus robuste : l'expansion de référence.

4.2.1 Appariement

La méthode communément admise pour appairer les mots-clés prédits et les mots-clés de référence est de les mettre en correspondance de manière exacte. Cet appariement, bien que simple à mettre en place, ne permet pas de prendre en compte certaines variantes telles que :

— les variantes flexionnelles : *réseau de neurones* et *réseaux de neurones* ;

— les variantes morpho-syntaxiques : *réseau de neurones* et *réseau neuronal* ;

— les synonymes : *rite funéraire* et *pratiques funéraires* ;

— les acronymes : *SVM* et *support vector machines*.

Le traitement de ces variantes peut être complexe à mettre en œuvre. C'est pourquoi seule la racinisation (Porter, 1980) est couramment employée : elle permet de traiter

une partie des variantes flexionnelles et morphologiques. Ce traitement est bien adapté à l'anglais mais moins au français où le phénomène d'allomorphie est courant. En français, la racinisation peut créer de fausses correspondances entre deux mots-clés n'ayant pas de rapport s'ils contiennent des allomorphes. Les allomorphes sont des mots qui ont une racine commune mais qui ne partagent pas de sens. Par exemple *empirique* et *empire* seront racinisés en *empir* : ils partagent une racine mais pas de sens.

4.2.2 Métriques

Les appariements obtenus précédemment sont utilisés par des métriques pour calculer des scores. Les métriques les plus utilisées sont la précision, le rappel et la F-mesure qui sont calculées grâce aux n meilleurs mots-clés (cf. équations 4.1, 4.2 et 4.3). La précision évalue le nombre de mots-clés corrects par rapport au nombre de mots-clés prédits par la méthode, tandis que le rappel évalue le nombre de mots-clés corrects par rapport au nombre de mots-clés de référence. La F-mesure correspond à la moyenne harmonique de ces deux valeurs. Dans les équations de cette section nous utiliserons Y pour désigner les mots-clés de référence, $\hat{Y}$ pour les mots-clés prédits, $\hat{Y}_{:n}$ pour les n meilleurs mots-clés prédits et $\hat{Y}_{:n} \bigcap Y$ représente l'intersection entre les mots-clés prédits et les mots-clés de référence.

$$P@n = \frac{|\hat{Y}_{:n} \bigcap Y|}{|\hat{Y}_{:n}|} \tag{4.1}$$

$$R@n = \frac{|\hat{Y}_{:n} \bigcap Y|}{|Y|} \tag{4.2}$$

$$F@n = \frac{2 * P@n * R@n}{P@n + R@n} \tag{4.3}$$

D'autres métriques, qui permettent de prendre en compte la qualité de l'ordonnancement des mots-clés, sont aussi utilisées. La MAP (mean Average Precision) représente la moyenne des précisions à chaque rang. Nous présentons sa formule dans l'équation 4.4. Elle rend compte de la capacité de la méthode à proposer les mots-clés corrects dans les premiers rangs. Cette mesure est utilisée entre autre par Basaldella et al. (2016); Boudin (2018); Gallina et al. (2020).

$$\text{MAP}(d) = \frac{1}{|M_d|} \sum_{i=1}^{|M_d|} \frac{|R_{d[:i]}|}{|M_{d[:i]}|} \tag{4.4}$$

Ensuite, la MRR (Mean Reciprocal Rank) calcule le rang du premier mot-clé correct. Dans l'équation 4.5, $rank_r$ représente le rang du mot-clé r. Cette mesure est utilisée par

Liu et al. (2010).

$$\text{MRR}(d) = \frac{1}{min\{rank_r, r \in R_d\}} \tag{4.5}$$

La Bpref (Binary Preference) pénalise les mots-clés corrects classés après des mots-clés incorrects. Dans l'équation 4.6, R_d représente l'ensemble des mots-clés corrects pour le document d et M_d l'ensemble des mots-clés prédits. Cette mesure est utilisée par Liu et al. (2010).

$$\text{Bpref}(d) = \frac{1}{|R_d|} \sum_{r \in R_d} 1 - \frac{|n \text{ mieux classés que } r|}{|M_d|} \tag{4.6}$$

Le NDCG (Normalized Discounted Cumulative Gain) calcule l'« importance » des mots-clés corrects en fonction de leur rang. Dans l'équation 4.7, DCG_i est le DCG optimal, c'est-à-dire l'ordonnancement qui positionne tous les mots-clés corrects avant les mots-clés incorrects. Cette mesure est utilisée par Marujo et al. (2012); Chen et al. (2018).

$$\begin{aligned} \text{DCG}(d) &= \frac{1}{|R_d|} \sum_{r \in R_d} \frac{1}{log_2(rank_r + 1)} \\ \text{NDCG}(d) &= \frac{\text{DCG}(d)}{\text{DCG}_i(d)} \end{aligned} \tag{4.7}$$

Toutes ces métriques, même les métriques ensemblistes, ne considèrent qu'un sous-ensemble de mots-clés prédits : les n meilleurs. Se restreindre aux n meilleurs mots-clés permet de comparer des méthodes qui peuvent proposer un nombre de mots-clés différent. Par exemple, les méthodes extractives sont sensibles à la longueur du document : elles proposent un nombre de mots-clés limité dans le cadre de notices scientifiques ou un nombre très important pour des articles scientifiques. Le choix de ce n n'est pas un problème résolu 2.3.3, il est généralement fixé de manière arbitraire. Pour éviter de le choisir, (Yuan et al., 2020) propose deux nouvelles métriques :

— la F@M (Modèle) qui remplace n par le nombre de mots-clés produits par la méthode ($|\hat{Y}|$). La F@M évalue donc la capacité d'une méthode à produire le nombre de mots-clés prédits qui correspond au nombre de mots-clés de référence ;

— la F@O (Oracle) qui remplace n par le nombre de mots-clés de référence ($|Y|$). La F@O évalue donc la capacité de la méthode à bien classer les mots-clés prédits.

En pratique, le nombre n de mots-clés adoptés pour évaluer les méthodes est généralement de 5 ou 10, ce qui correspond au nombre moyen de mots-clés annotés respectivement par les auteurs et les indexeurs. Nous montrons dans notre évaluation extrinsèque (cf. chapitre 7), qu'un nombre de 4 ou 5 mots-clés prédits est un bon compromis pour la tâche de recherche d'informations.

4.2.3 Expansion de référence

Pour rendre l'évaluation plus robuste aux différentes variantes de mots-clés, décrites dans la section 4.2.1, des techniques sont proposées pour compléter les mots-clés de référence de manière automatique. Chan et al. (2019) propose d'augmenter la référence en ajoutant des variantes de mots-clés : acronymes et synonymes. Leur étude des mots-clés de référence montre que certains mots-clés de référence contiennent un acronyme, par exemple *support vector machine (svm)*. Pour extraire ces acronymes entre parenthèses et les ajouter à la référence, ils utilisent des expressions régulières. Pour trouver des synonymes de mots-clés, ils exploitent la fonction de redirection automatique de Wikipédia :

— si la recherche du mot-clé *solid state disk* mène sur l'article *solid state drive* ; alors *solid state disk* est considéré comme un synonyme de *solid state drive*.

— Si la recherche du mot-clé *ssd* mène sur une page de désambiguïsation qui propose pour chaque sens de cet acronyme une forme étendue, et qu'une de ces formes étendues apparaît dans le document (par exemple *solid state disk*) ; alors cette forme étendue est considérée comme un synonyme de *ssd*.

Leurs expériences montrent que ces heuristiques ajoutent au moins une variante pour 14,1 % des mots-clés de référence et qu'elles sont à l'origine d'une augmentation de 0,1 de F@M lors de l'évaluation de méthodes de génération de mots-clés.

Nous pouvons encore citer Kim et al. (2010) qui ajoutent des variantes syntaxiques aux mots-clés. Ils se concentrent sur les mots-clés sous forme génitive. Par exemple *policy of school* est la version génitive de *school policy*. Ainsi, ils génèrent et ajoutent comme variante d'une forme génitive, la forme canonique correspondante.

Dans le jeu de données KPTimes, certains mots-clés sont associés à des variantes. En effet, les mots-clés de référence proviennent d'un vocabulaire contrôlé, dans lequel certains mots-clés sont associés à des termes apparentés. Par exemple, des acronymes sont associés à leurs formes expansées ; pour les artistes le nom d'usage est associé à leur nom de scène (« Abel Tesfaye » et « The Weeknd ») ; ou encore une appellation populaire est associée à son appellation officielle (« Obamacare » et « Affordable Care Act »).

4.3 Conclusion

Nous avons présenté le cadre expérimental de la tâche de production automatique de mots-clés : d'abord, les jeux de données utilisés dans nos expérimentations, présentées dans les prochains chapitres ; ensuite le processus d'évaluation des méthodes de production automatique de mots-clés.

Ainsi, les jeux de données que nous utilisons couvrent un large spectre de genres, de types et de tailles de documents ainsi que de types d'annotation. Ces jeux de données sont aussi les plus utilisés par la communauté scientifique. Cet ensemble de jeux de données comporte principalement des documents scientifiques, qui sont notre objet d'étude mais il

comporte aussi des articles journalistiques qui nous servent à étudier la généralisation des méthodes. Ces jeux de données contiennent des documents longs : les articles scientifiques complets qui comportent en moyenne $\simeq$ 8500 mots ; et des documents courts : les notices scientifiques qui contiennent $\simeq$ 150 mots en moyenne. Ces jeux de données sont annotés par différents annotateurs, non-professionnels pour la plupart. Nous avons souligné l'hétérogénéité des processus d'annotation en mots-clés, en particulier pour les mots-clés lecteurs. La qualité de ces annotations non-professionnelles est donc très variable. Il est à noter que la majorité de ces jeux de données est en anglais, même s'il existe quelques initiatives en français et dans d'autres langues. Cette prépondérance de l'anglais s'explique par la place centrale de cette langue dans la communauté scientifique et par son nombre de locuteurs. Les documents disponibles librement et annotés en mots-clés sont rares et proviennent généralement des mêmes sources. C'est pourquoi certains documents apparaissent dans plusieurs jeux de données (SemEval-2010 est composé à 84 % de document de KP20k).

Nous avons également présenté le processus d'évaluation des méthodes de production automatique de mots-clés. Ce processus apparie les mots-clés prédits à une référence puis calcule un score. L'appariement entre les mots-clés prédits et les mots-clés de référence s'effectue grâce à une comparaison exacte qui ne permet pas de prendre en compte les variantes des mots-clés. Pour assouplir cette comparaison, un algorithme de racinisation est systématiquement utilisé. Avec ces appariements, des scores sont calculés grâce aux métriques ensemblistes classiques (précision, rappel, F-mesure) et aux métriques d'ordonnancement de la recherche d'informations (MAP). Les métriques les plus rapportées dans les travaux sont le rappel et la F-mesure. Pour compléter les références et rendre plus robuste l'appariement entre les mots-clés de référence et les mots-clés prédits, certains jeux de données incluent des variantes de mots-clés de référence telles que la version expansée d'un acronyme ou un synonyme. Cet ajout de variantes s'effectue de manière manuelle grâce à une expertise linguistique ou de manière automatique avec des ressources externes.

Chapitre 5

KPTimes : des mots-clés éditeurs pour la génération de mots-clés

Dans le chapitre 4, nous avons recensé les principaux jeux de données exploités pour la production de mots-clés. Ces jeux de données contiennent en moyenne 1 000 documents, à l'exception de KP20k ; cette quantité n'est pas suffisante pour entraîner des méthodes neuronales génératives qui requièrent des centaines de milliers de documents. Ainsi, il manque un jeu de données similaire à KP20k (c'est-à-dire de grande taille) mais d'un domaine différent pour pouvoir évaluer la généralisation des méthodes de production automatique de mots-clés. Nous avons retenu le domaine journalistique car de grandes quantités de documents sont facilement collectables et ils sont parfois annotés en mots-clés. De plus, les mots-clés des documents scientifiques sont majoritairement annotés par les auteurs. Cette annotation génère de nombreux biais que nous détaillerons au chapitre 6. En particulier, les effets de mode biaisent le choix des mots-clés, l'objectivité des annotateurs et la cohérence entre les annotateurs.

Pour pallier ces inconvénients, nous avons constitué le jeu de données KPTimes, composé d'articles journalistiques en anglais et suffisamment grand pour entraîner des méthodes neuronales. KPTimes documente le domaine, la provenance et les auteurs de chaque document grâce à des métadonnées riches, contrairement à KP20k dont l'origine des documents est incertaine, et dont les différents domaines ne sont pas explicités. L'annotation en mots-clés a suivi un processus rigoureux qui garantit leur qualité et la cohérence de l'annotation.

Dans ce chapitre, nous présentons d'abord le processus de collecte de KPTimes. Nous présentons ensuite ses statistiques en le comparant aux jeux de données comparables en termes de genre (articles journalistiques) et en termes de taille (KP20k). Enfin, nous analyserons les performances des méthodes de production automatique de mots-clés sur KPTimes ainsi que les capacités de généralisation des méthodes neuronales sur KPTimes et KP20k.

5.1 Constitution du jeu de données

Dans cette section nous présentons d'abord le processus de constitution du jeu de données KPTimes : la collecte des données et le filtrage des documents. Ensuite, nous décrivons ce jeu de données en exposant ses statistiques. Un exemple de document extrait de KPTimes est présenté dans la figure 5.1.

> **Muslim Women in Hijab Break Barriers : 'Take the Good With the Bad'**
>
> When Ginella Massa, a Toronto-based TV reporter, recently accepted a request to host an evening newscast, she was not planning or expecting to make history for wearing a hijab. She was just covering for a colleague who wanted to go to a hockey game. And that's how Ms. Massa, who works at CityNews in Toronto, became the first Canadian woman to host a newscast from a large media company while wearing the head scarf. [...] This new trend of inclusion occurs amid a more sinister one, as reported hate crimes against Muslims are on the rise in the United States and Canada. The F.B.I. says that a surge in hate crimes against Muslims has led to an overall increase in hate crimes in the United States ; Muslims have borne the brunt of the increase with 257 recorded attacks. [...] In Canada, where Ms. Massa has lived since she was a year old, the number of reported hate crimes has dropped slightly overall, but the number of recorded attacks against Muslims has grown : 99 attacks were reported in 2014, according to an analysis by the news site Global News of data from Statistics Canada, a government agency. [...]
>
> **Mots-clés de référénce :** US ; Islam ; Fashion ; Muslim Veiling ; Women and Girls ; (News media, journalism) ; Hate crime ; Canada

FIGURE 5.1 – Document extrait de KPTimes (id : `ny0296216`). Les mots apparaissant dans le document sont colorés.

5.1.1 Sélection des sources de données

Nous avons constitué ce jeu de données de manière automatique à partir d'articles journalistiques extraits du site `nytimes.com`. Ce site a été choisi car les métadonnées des articles contiennent des mots-clés annotés de manière semi-automatique par les éditeurs du journal. Les éditeurs se trouvent à la fin de la chaîne de production des articles et sont chargés des dernières vérifications avant publication : choisir le titre, les mots-clés et mettre en page le journal.

L'annotation éditeur se fait de manière semi-automatique. Pour cela un système d'indexation contrôlée propose un ensemble de mots-clés aux éditeurs. Ceux-ci peuvent alors réviser l'ensemble, c'est-à-dire ajouter ou supprimer des mots-clés, qu'ils fassent partie du vocabulaire contrôlée ou non. [1] Ces révisions sont ensuite prises en compte par le système d'indexation contrôlée pour les prochaines indexations. Les mots-clés du système d'indexation contrôlée relèvent de cinq catégories (Sandhaus, 2008) :

— lieux, p. ex. *Brooklyn*,

1. `https://media.lac-group.com/blog/rules-based-tagging-metadata`

— personnes, p. ex. *Bernie Sanders*,

— titres d'oeuvres, p. ex. *Snow White and the Seven Dwarfs*,

— structures et entreprises, p. ex. *Pfizer Inc*,

— sujets généraux, p. ex. *Skateboarding*.

Ainsi ce processus d'annotation est mixte : il combine les avantages de l'annotation contrôlée (cohérence) grâce à l'algorithme, et libre (exhaustivité) grâce à la révision des éditeurs.

Dans le but d'observer les capacités de généralisation des méthodes à d'autres types d'annotation en mots-clés, nous constituons un second ensemble d'articles à partir du site `japantimes.co.jp` qui sera intégré à l'ensemble de test de KPTimes. Les articles du Japan Times contiennent aussi des mots-clés dans les métadonnées. Depuis 2013, l'édition du week-end du Japan Times inclut l'*International New York Times*, ce qui laisse penser que ces deux journaux ont une ligne éditoriale similaire et qu'ils traitent de sujets similaires. Nous décrivons séparément ces deux sous-ensembles d'articles extraits du New York Times et du Japan Times et le jeu de données KPTimes dans son ensemble.

Après filtrage (voir section 5.1.3), les documents sont séparés en ensembles de test, entraînement et validation. L'assignation des documents à ces ensembles se fait de manière aléatoire, en vérifiant que le ratio de mots-clés présents, la longueur et la distribution dans les différentes catégories soient comparables. L'ensemble de test est composé par les documents du New York Times et du Japan Times en quantités égales. Nous mettons à disposition les documents au format `jsonl`. [2]

5.1.2 Collecte des données

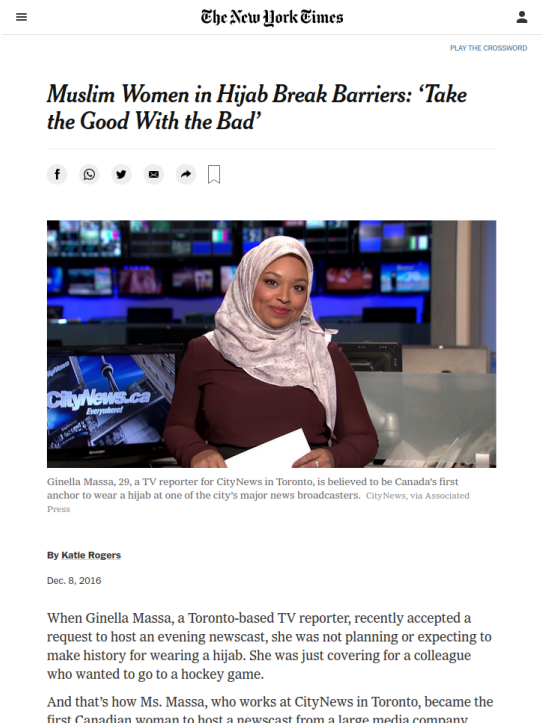
La collecte des données concerne deux journaux en anglais : le New York Times et le Japan Times.

Les articles du New York Times ont été collectés de manière automatique par moissonnage. Les articles disponibles gratuitement sont répertoriés par date sur une page web [3], nous en avons extrait les URL des articles de 2006 à 2017.

Cela représente 296 974 articles au format `html` composés du titre, de la *headline* (résumé en une ligne), du corps de l'article et des métadonnées. Parmi les métadonnées (voir figure 5.2b), nous conservons la date de publication, la catégorie de l'article, l'auteur et les mots-clés. La mise en page de l'article (décrite par le format `html`) encode des annotations telles que la séparation en paragraphes, les ancres de liens et la mise en forme matérielle (voir figure 5.2a). Nous avons choisi de ne pas incorporer ces informations dans KPTimes mais nous indiquons l'URL de chaque document qui permet de reconstruire la collection et d'accéder aux documents `html` originaux.

Les articles du Japan Times sont au nombre de 11 057 et ont été collectés dans une période de 2008 à 2019 (dont 70 % datent de 2019). Les métadonnées sont similaires à

2. https://github.com/ygorg/KPTimes
3. https://spiderbites.nytimes.com/



(a) Capture d'écran d'un article sur le site du New York Times.

```
<html><body>
<title>Muslim Women in Hijab Break Barriers:
'Take the Good With the Bad' - The New York
Times</title>
<meta name="description" content="
    Even as reports of hate crimes against
    Muslims rise in America and Canada,
    hijabis are appearing in makeup ads,
    beauty pageants and news anchor chairs.
    ">
<meta name="byl" content="By Katie Rogers">
<meta name="news_keywords" content="
    Muslim Veiling,,Hate crime, Women and
    Girls,Canada,US,Islam,Fashion,
    News media;journalism,">
<meta name="CG" content="world">
<meta name="SCG" content="americas">
<meta name="pdate" content="20161208">
<meta name="url" content="
    https://www.nytimes.com/2016/12/08/
    world/americas/hijab-muslim-women.html">
...
</head><body>
When Ginella Massa, a Toronto-based ...
</body></html>
```

(b) Code source d'un article du site du New York Times.

FIGURE 5.2 – Version web et `html` de l'article `ny0296216` du jeu de données KPTimes.

celles du New York Times : les articles sont séparés en catégories, sont associés à des mots-clés et contiennent une *headline*.

5.1.3 Filtrage des documents collectés

Un examen manuel des documents montre que certains articles sont redondants, trop longs ou trop courts. Nous décrivons ici le processus de filtrage de ces articles.

Redondance Deux articles sont des doublons s'ils partagent le même titre et le même texte, auquel cas nous n'en conservons qu'un. La suppression des doublons concerne 2 233 articles. Les cas de doublons arrivent lorsque le même article est publié à deux dates différentes.

Deux types d'articles présentent aussi des similarités sans être des doublons :

— les articles récurrents, qui partagent un même contenu, tels que l'agenda des sorties ou les résultats du loto, et qui ne diffèrent que par des dates, des chiffres et autres informations ne modifiant pas les sujets abordés ;

— les articles suivis, couvrant un même évènement dans le temps.

Ces deux types d'articles ont la particularité de partager un identifiant journalier (*slug*) et le même ensemble de mots-clés. Par exemple, l'identifiant journalier des 464 articles

présentant les résultats de la loterie est « lottery-numbers-for-new-york-new-jersey-and-connecticut ». Sur ces 464 articles 296 partagent le même ensemble de mots-clés : *Connecticut*, *Lotteries*, *New Jersey*, *New York State*. Pour supprimer les articles similaires, nous les regroupons à l'aide de leur identifiant journalier, puis nous supprimons les groupes d'articles où plus de 70 % des documents ont le même ensemble de mots-clés, soit 2 533 articles. Nous avons fixé ce ratio à 70 % pour inclure certains articles similaires dont le contenu diffère significativement malgré une majorité de mots-clés identiques. Ce ratio permet par exemple de conserver une série d'articles couvrant des manifestations en Thaïlande qui partagent le même identifiant journalier « thailand-protest » et des mots-clés similaire (Bangkok, Thailand, Yingluck Shinawatra, Thaksin Shinawatra) mais qui ont chacun un sujet différent.

Longueur Nous souhaitons homogénéiser la taille des documents en écartant les documents trop longs qui nécessitent beaucoup de mémoire pour être traités avec des méthodes neuronales, et les documents trop courts qui contiennent peu d'informations. Les articles trop longs sont identifiés à l'aide de l'écart interquartile utilisé pour identifier les valeurs extrêmes en termes de longueur de mots. Pour la limite haute, nous choisissons 10 fois l'écart interquartile. Nous supprimons ainsi 77 articles dont la longueur est supérieure à 8 163 (le plus long fait 31 893 mots). Ces articles sont des reportages ou des transcriptions de discours, ils diffèrent donc des articles journalistiques. Les documents ayant une longueur inférieure à 100 mots sont aussi supprimés, soit 9 326 documents. Ces documents correspondent pour les 2/3 à des brèves de la catégorie sport.

Les documents du Japan Times sont filtrés de la même manière que les articles provenant du New York Times pour la redondance. En termes de longueur seuls les documents supérieurs à 1 fois l'écart interquartile sont supprimés, les documents du Japan Times étant en moyenne plus courts que ceux du New York Times.

5.1.4 Description statistique

Le tableau 5.1 présente les statistiques des ensembles de test de KPTimes (qui rassemble NYTimes et JPTimes), ainsi qu'à des fins de comparaison, les jeux de données d'articles journalistiques KPCrowd et DUC-2001 et le jeu de données de notices scientifiques de grande taille KP20k. Nous détaillons et commentons ci-après ces statistiques.

Taille du jeu de données KP20k contient deux fois plus de documents d'entraînement que KPTimes, 530 K contre 260 K, mais le même nombre de documents de test et de validation (20 K). Malgré sa taille moins importante par rapport à KP20k, KPTimes permet néanmoins l'entraînement de méthodes neuronales. Les tailles des ensembles de test de KPCrowd et DUC-2001 (centaine de documents) ne sont pas dans le même ordre de grandeur que KPTimes et KP20k (de l'ordre de dizaines de milliers). DUC-2001 ne propose pas de documents d'entraînement.

		Corpus			Document			Mots-clés	
Corpus	**Ann.**	**#Entr.**	**#Val.**	**#Test**	**#mots**	**#mc**	**%abs**	**#uniq.**	**#ass.**
KPTimes	*E*	260 K	20 K	20 K	738	5,0	38,4	20 535	5,0
JPTimes	*E*	-	-	10 K	570	5,0	24,2	8 611	5,9
NYTimes	*E*	260 K	20 K	10 K	905	5,0	52,5	13 387	3,8
KPCrowd	*L*	450	-	50	465	46,2	8,1	1 937	1,2
DUC-2001	*L*	-	-	308	847	8,1	3,1	1 800	1,4
KP20k	*A*	530 K	20 K	20 K	176	5,3	42,4	53 489	2,0

TABLEAU 5.1 – Comparaison des statistiques de KPTimes et ses sous-ensembles de test JPTimes et NYTimes avec les jeux de données d'articles journalistiques et KP20k. Les mots-clés de référence sont annotés par des *L*ecteurs, des *E*diteurs ou des *A*uteurs. La table présente le nombre de documents dans les corpus d'entraînement (#Entr.), de validation (#Val.) et de test (#Test) ainsi que le nombre moyen de mots (#mots), de mots-clés (#mc) et le ratio de mots-clés absents (%abs) par document. Les colonnes #uniq. et #ass. montrent le nombre de mots-clés uniques et le nombre moyen d'assignation d'un mot-clé.

Nombre de mots-clés Le jeu de données KPTimes est assez proche de KP20k en termes de nombre de mots-clés malgré les genres différents de documents qu'ils rassemblent et les différentes procédures d'annotation. Ils ont respectivement 5,3 et 5,0 mots-clés par document en moyenne. Les jeux de données journalistiques KPCrowd et DUC-2001 annotés par des lecteurs contiennent plus de mots-clés en moyenne que KPTimes et KP20k, respectivement, 46,2 et 8,1.

Mots-clés absents KPCrowd et DUC-2001 ont moins de 10 % de mots-clés absents, cela est lié à leur annotation lecteur qui privilégie les mots-clés présents dans le document à l'inverse de l'annotation éditeur de KPTimes. L'ensemble de test de KPTimes est similaire à KP20k en termes de mots-clés absents, respectivement 38,4 % et 42,4 %. Les ensembles d'apprentissage de KPTimes (documents du NYTimes) et KP20k ont un taux de mots-clés absents du même ordre de grandeur, respectivement 52,5 % et 42,4 %.

Qualité d'annotation KPTimes diffère des autres jeux de données par son annotation. L'annotation éditeur de KPTimes est semi-automatique, la proposition des mots-clés par un système automatique permet d'associer à un concept le même mot-clé de manière plus cohérente qu'une annotation libre lecteur ou auteur. Cette différence d'annotation est mise en lumière par le pourcentage de mots-clés en fonction du nombre d'assignations par document. La figure 5.3 donne ce pourcentage pour les ensembles de test des trois jeux de données KPCrowd, KP20k et KPTimes annotés respectivement par des lecteurs, des auteurs et des éditeurs. Par exemple, pour KP20k, 80 % des mots-clés ne sont assignés qu'à un document, 10 % des mots-clés à deux documents, 3 % des mots-clés à trois documents, etc. Entre l'annotation lecteur et l'annotation éditeur, le nombre de mots-clés assigné une seule fois baisse de 22,6 %. Ce chiffre révèle la cohérence de l'annotation aidée d'un

vocabulaire contrôlé par rapport à une annotation libre où le même concept peut être associé à différentes variantes de mots-clés (voir figure 2.2).

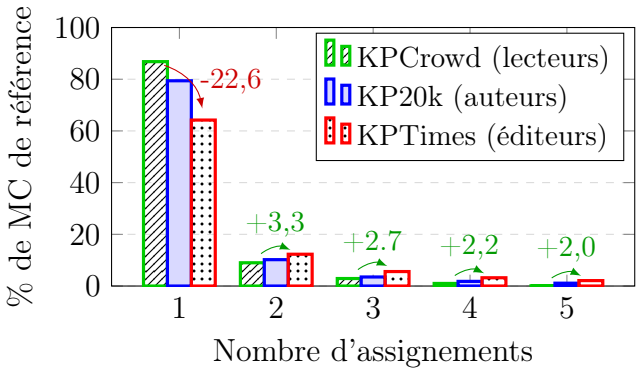


FIGURE 5.3 – Distributions de l'assignement des mots-clés de référence pour les mots-clés assignés à 5 documents ou moins.

Le tableau 5.2 précise le nombre moyen d'assignations d'un mot-clé à un document pour les ensembles de test, d'apprentissage et de validation des jeux de données KP20k et KPTimes. L'augmentation du nombre moyen d'assignations d'un mot-clé dans l'ensemble d'entraînement par rapport à l'ensemble de test s'explique par le nombre plus important des documents. Un modèle entraîné sur KP20k apprend ≃650 000 mots-clés différents alors qu'un modèle entraîné sur KPTimes en apprend seulement ≃100 000. L'espace de recherche plus restreint de KPTimes facilite donc l'apprentissage de modèles supervisés. La cohérence dans l'annotation de KPTimes décrite plus haut fait qu'un même mot-clé est présenté en moyenne 13 fois au modèle pendant l'apprentissage, contre 4 fois pour KP20k. Cette meilleure représentativité des mots-clés de KPTimes par rapport à KP20k nous porte à croire qu'elle devrait influencer favorablement les performances des modèles appris.

	KP20k		**KPTimes**	
	#ass.	#uniq.	#ass.	#uniq.
Test	2,0	53 489	5,0	20 535
Val.	2,0	52 975	3,9	13 269
Entr.	4,3	648 697	13,1	101 748

TABLEAU 5.2 – Comparaison du nombre de mots-clés total et du nombre moyen d'assignation d'un mot-clé entre les ensembles de test, de validation et d'entraînement de KPTimes et KP20k.

Catégories des documents Les articles journalistiques de KPTimes sont classés en plusieurs catégories et sous-catégories thématiques (par exemple *Science*, *Région de NY* et *Sports*). Ces catégories peuvent servir à entraîner des modèles spécialisés ou à analyser plus finement les performances des méthodes de production automatique de mots-clés. Le tableau 5.3 présente la distribution des articles dans chacune des catégories des sous-ensembles de test NYTimes et JPTimes. Les catégories de ces deux sous-ensembles ne

sont pas alignées. En effet, il n'y a pas de lien direct entre les deux ensembles de catégories bien que la catégorie *Sports* se retrouve dans les sous-catégories de JPTimes. Cette absence d'alignement s'explique par la différence de granularité des deux ensembles de catégories thématiques, NYTimes regroupe les articles par thématiques alors que JPTimes les regroupe plutôt par aire géographique. Seule la catégorie « Business » est comparable en termes de nombre de documents entre ces deux sous-ensembles.

Catégorie	% d'articles
Sports	24,0
Monde	19,7
Business	19,1
Région de NY	15,3
U.S.	15,3
Technologie	3,4
Science	3,0
Politique	0,1

(a) NYTimes

Catégorie	% d'articles
National	35,1
Monde	28,1
Business	23,0
Asie-Pacifique	12,3
Reference	1,5
General	0,1

(b) JPTimes

TABLEAU 5.3 – Distribution des articles du jeu de données KPTimes par catégories dans l'ensemble de test.

5.2 Performances du jeu de données

Dans cette section, nous présentons et comparons les performances de plusieurs méthodes de production de mots-clés pour, dans un premier temps, attester de la validité de KPTimes, et dans un second temps, étudier la généralisation des méthodes à différents genres de documents et types d'annotations.

Nous comparons tout d'abord les résultats obtenus sur les jeux de données d'articles journalistiques (du même genre que KPTimes) mais dont l'annotation en mots-clés diffère. Ensuite, nous comparons les performances des méthodes sur les deux sous-ensembles de test de KPTimes : NYTimes et JPTimes. Ces ensembles de test comportent des documents du même genre et une annotation en mots-clés similaire. Enfin, nous analysons la capacité de généralisation de la méthode CopyRNN grâce à une évaluation croisée sur KPTimes et KP20k, dont le genre des documents et le type d'annotation en mots-clés diffèrent.

Nous choisissons de comparer les méthodes suivantes (décrites dans les sections 2.3 et 3.2) : FirstPhrases, TF×IDF, MultipartiteRank, Kea et CopyRNN. Nous rappelons les principales différences : FirstPhrases utilise uniquement la position ; TF×IDF la fréquence et la spécificité à un document ; Kea combine ces deux indicateurs de manière supervisée ; MultipartiteRank utilise la centralité dans un graphe ; CopyRNN est une méthode neuronale générative. Nous utiliserons **CopyNews** pour désigner la méthode CopyRNN en-

traînée sur les articles journalistiques de KPTimes, et **CopySci** pour désigner CopyRNN entraînée sur les articles scientifiques de KP20k. Les paramètres utilisés sont ceux décrits par Meng et al. (2017). Les mots-clés produits automatiquement sont évalués grâce au protocole décrit dans la section 4.2.

5.2.1 Comparaison aux jeux de données journalistiques

	KPTimes		**DUC-2001**		**KPCrowd**	
	F@10	MAP	F@10	MAP	F@10	MAP
FirstPhrases	11,4	10,7	24,3	22,3	17,1	16,5
TF×IDF	12,4	12,8	23,0	21,6	16,9	15,8
MPRank	14,0	13,3	25,3	**24,9**	**18,2**	**17,0**
Kea	13,8	14,1	**26,2**	24,5	17,3	16,7
CopyNews	**31,9**	**38,7**	10,5	7,2	8,4	4,2

TABLEAU 5.4 – Performances de différents modèles de production de mot-clés sur des jeux de données d'articles journalistiques.

Le tableau 5.4 compare les performances de différentes méthodes de production de mots-clés sur des jeux de données journalistiques : KPTimes, DUC-2001 et KPCrowd. Les performances des méthodes extractives (FirstPhrases, TF×IDF et MPRank) sont bien plus basses sur KPTimes que sur DUC-2001 et KPCrowd : 10 points de différence de MAP entre KPTimes et DUC-2001. Cette grande différence s'explique par leurs proportions différentes de mots-clés absents : KPTimes en contient 38,4 % et DUC-2001 3,1 %. En effet, les méthodes extractives ne proposent que des mots-clés présents dans les documents, c'est-à-dire que 38,4 % des mots-clés de KPTimes ne peuvent être retournés par ces méthodes.

Les performances de la méthode CopyNews (CopyRNN entraîné sur KPTimes) sont étonnamment faibles sur DUC-2001 et KPCrowd (respectivement 7 % et 4 % de MAP) et sont bien inférieures aux performances des méthodes extractives. Cette grande différence est liée aux mots-clés de référence utilisés pour l'évaluation : CopyNews est entraîné à reproduire les mots-clés éditeurs de KPTimes alors que les mots-clés de DUC-2001 et KPCrowd ont été annotés par des lecteurs. Les mots-clés de ces deux types d'annotation sont très différents, en effet, seulement 13 % des mots-clés de DUC-2001 et 19 % des mots-clés de KPCrowd apparaissent aussi dans KPTimes. [4]

Pour tempérer ces faibles résultats nous présentons dans la figure 5.4 un document extrait de DUC-2001 ainsi que ses mots-clés de référence et ceux produits par CopyNews. Même si aucun des mots-clés de CopyNews n'est commun à ceux de la référence, ces mots-clés sont (pour la plupart) pertinent et reflètent les spécificités de l'annotation éditeur. Par rapport à la référence, les sujets principaux sont bien identifiés (la tuberculose et la

4. Ces statistiques ont été calculées sur les ensembles de test.

The tuberculosis rate in U.S. prisons may be more than three times higher than on the outside, federal health officials said Thursday, urging testing, isolation and other measures to curb TB behind bars. [...] In New Jersey, inmates had a TB rate of 110 per 100,000 in 1987, 11 times higher than the general New Jersey population. In California, the rate was nearly six times higher – 80 per 100,000. Tuberculosis, a contagious, bacterial lung disease, occurs in about 22,000 new cases each year in the United States ; most can be cured with drug treatment. As many as 7 percent of Americans have latent TB infections, and about 10 percent of them will someday develop a case of tuberculosis itself. « Persons at highest risk... are close contacts, » the CDC said, noting that TB can pose particular problems in prisons, where there is often overcrowding and « where the environment is often conducive to airborne transmission of infection among inmates, staff and visitors. » [...] The spread of AIDS-virus infections may play a part in the spread of TB in prisons, the CDC said. AIDS weakens the immune system, making patients susceptible to infections other people might ward off, including tuberculosis. AIDS tests should be offered to all inmates with known TB infections, the CDC report said.

CopyNews : tuberculosis, new jersey, us, medicine and health, prisons
Mots-clés de référence : tuberculosis rate, u.s. prisons, aids-virus infections, airborne transmission, tuberculosis cases, cdc

FIGURE 5.4 – Exemple de document de l'ensemble de test du corpus DUC-2001 (id : AP890511-0126)).

prison), le lieu est aussi bien identifié (« us ») même s'il s'agit d'une variante du mot-clé de référence. Le modèle prédit aussi le mot-clé « *new jersey* » qui est mentionné dans l'article mais n'est pas pertinent dans ce contexte. Le mot-clé généré « *medicine and health* » est intéressant car il est spécifique à l'annotation éditeur. Il décrit le sujet global du document et a plus vocation à catégoriser le document qu'à décrire un de ses sujets en particulier. Le modèle n'identifie en revanche pas les concepts de « cdc » (*Centers for Disease Control*) qui est la source de cet article, de « aids-virus infection » qui facilite la transmission de la tuberculose et de « *airborne transmission* » qui décrit la méthode de propagation du virus. Ainsi, les mots-clés générés par le modèle, malgré leur faible correspondance aux mots-clés de référence, sont cohérents avec l'annotation éditeur qui identifie le lieu, les concepts principaux et les acteurs (individus, entreprises, etc.).

5.2.2 Généralisation des méthodes neuronales

Nous présentons les performances des méthodes de production de mots-clés sur KPTimes et ses deux sous-ensembles de test, NYTimes et JPTimes dans le tableau 5.5. Les méthodes extractives ont des performances plus élevées sur JPTimes que sur NYTimes : la F@10 de Kea est de 11,0 sur NYTimes contre 16,6 sur JPTimes. Cette différence s'explique par le fait que JPTimes contient moins de mots-clés absents (24,2 %) que NYTimes (52,5 %). Là encore, le taux de mots-clés absents pénalise les méthodes exclusivement extractives. Le modèle CopyNews entraîné sur les documents du NYTimes, est moins performant sur JPTimes, le score de MAP est divisé par deux. Ces constatations montrent combien le modèle est dépendant de la référence sur laquelle il a été entraîné et c'est d'autant plus vrai pour les mots-clés absents.

Le tableau 5.6 présente les performances de CopyNews et CopySci testées de manière croisée sur KPTimes et KP20k. Les performances hors situation de généralisation, c'est-

	(NYT + JPT) **KPTimes**		**NYTimes**		**JPTimes**	
	F@10	MAP	F@10	MAP	F@10	MAP
FirstPhrases	11,4	10,7	9,2	8,4	13,5	13,1
TF×IDF	12,4	12,8	9,6	9,4	15,1	16,2
MPRank	14,0	13,3	11,2	10,1	16,8	16,5
Kea	13,8	14,1	11,0	10,8	16,6	17,5
CopyNews	**31,9**	**38,7**	**39,3**	**50,9**	**24,6**	**26,5**

TABLEAU 5.5 – Performances de différents modèles de production de mot-clés sur KPTimes et ses sous-ensembles de test.

à-dire avec CopyNews testé sur KPTimes et CopySci testé sur KP20k, sont plus élevées pour CopyNews que pour CopySci. Cette différence de performance s'explique par la plus grande cohérence des mots-clés de KPTimes. En situation de généralisation par contre, CopySci est plus performant que CopyNews. Les performances, par rapport à une évaluation non croisée, sont divisées par $\simeq 5$ pour CopyNews et $\simeq 2$ pour CopySci. Ainsi CopySci généralise donc mieux que CopyNews. Ce résultat est étonnant étant donné les performances plus élevées de CopyNews évalué sur KPTimes. Pour étudier plus finement cette différence de performances en situation de généralisation, nous nous intéressons au nombre de mots-clés générés par ces modèles.

	KPTimes		**KP20k**	
	F@10	MAP	F@10	MAP
CopyNews	**31,9**	**38,7**	6,6	5,1
CopySci	14,9	15,2	**25,5**	**28,9**

TABLEAU 5.6 – Performances de généralisation du modèle CopyRNN entraîné sur KPTimes et KP20k.

Le tableau 5.7 présente le pourcentage de mots-clés présents et de mots-clés absents produits par les modèles CopyNews et CopySci sur KPTimes et KP20k. Le pourcentage de mots-clés absents produits automatiquement par le modèle CopyNews (38,8 %) est très proche de celui de la référence pour KPTimes (38,4 %). Le pourcentage de mots-clés absents produits automatiquement par le modèle CopySci (8,0 %) est très éloigné de celui de la référence pour KP20k (42,4 %). Ces résultats montrent de nouveau la difficulté du modèle CopyRNN à reproduire l'annotation auteur, moins cohérente que l'annotation éditeur. Ils montrent aussi que le modèle CopyRNN possède la capacité de générer un grand nombre de mots-clés absents, ce que nous ne pouvions savoir sans le jeu de données KPTimes et son annotation éditeur. Nous remarquons que le pourcentage de mots-clés présents et absents produits par un modèle reste similaire en situation de généralisation, c'est-à-dire lorsque l'ensemble de test n'appartient pas au jeu de données utilisé pour l'entraînement. Dans cette situation de généralisation, le modèle CopySci sur KPTimes

obtient de meilleurs résultats que CopyNews sur KP20k. CopySci produit presque exclusivement des mots-clés présents (94,6 %) alors que CopyNews n'en produit qu'une courte majorité (51,1 %). La généralisation pour la production de mots-clés absents est plus difficile que la production de mots-clés présents. Généraliser à un discours scientifique, lorsque l'apprentissage a été réalisé sur un discours généraliste, handicape la génération des mots-clés. Par exemple, le modèle CopyNews assigne le mot-clé *nyc* (*New York City*) à 2 137 documents de KP20k. L'étude manuelle d'un échantillon de 30 documents montre que ce mot-clé n'a de rapport avec aucun de ces documents.

	KPTimes		**KP20k**	
	Prs	Abs	Prs	Abs
CopyNews	61,2	38,8	51,1	48,9
CopySci	94,0	5,1	92,0	8,0

TABLEAU 5.7 – Pourcentage de mots-clés présents et absents produit automatiquement par le modèle CopyRNN entraîné sur KPTimes (CopyNews) et KP20k (CopySci) et testé sur ces deux jeux de données.

5.3 Conclusion

Nous avons présenté notre contribution, le jeu de données KPTimes, composé d'articles journalistiques annotés de manière semi-supervisée par des éditeurs, et de taille suffisamment importante pour permettre l'entraînement de méthodes neuronales. Les documents ont été collectés de manière automatique sur les sites internet du New York Times et du Japan Times. Les documents sont annotés de manière semi-automatique grâce à un système d'indexation contrôlée, vérifiée et enrichie par les éditeurs. Ce processus d'annotation résulte ainsi en une annotation beaucoup plus cohérente, facilitant l'apprentissage des méthodes neuronales.

Nous avons ensuite mesuré les performances de plusieurs méthodes de production automatique de mots-clés sur KPTimes, puis nous les avons comparées à celles obtenues sur deux jeux de données journalistiques. Nous avons constaté que les méthodes extractives étaient moins performantes sur KPTimes que sur les autres jeux de données (ce qui est lié au nombre plus important de mots-clés absents). Nous avons aussi comparé les performances de méthodes neuronales entraînées sur KP20k et KPTimes, et avons trouvé que le modèle entraîné sur KPTimes est plus performant que celui entraîné sur KP20k. La création de KPTimes nous a permis de réaliser une première étude sur la capacité de généralisation des méthodes neuronales, c'est-à-dire lorsque l'entraînement et le test sont effectués sur des jeux de données différents. Cette étude a montré la meilleure capacité de généralisation du modèle neuronal CopyRNN lorsqu'il est entraîné sur le jeu de données scientifiques KP20k que sur le jeu de données généralistes KPTimes. Ce résultat est surprenant étant donné la meilleure qualité des mots-clés de KPTimes mais il est en fait

lié à la faible capacité du modèle entraîné sur KP20k à produire des mots-clés absents. En effet, il est moins risqué, pour avoir un rapport avec le document, de produire un mot-clé présent, limité au document, plutôt qu'un mot-clé absent. Nous allons dans le chapitre suivant poursuivre nos expérimentations des différentes méthodes extractives et génératives en nous plaçant dans un cadre expérimental strict.

Chapitre 6

Évaluation à large couverture

Dans le chapitre 5, nous avons introduit et évalué KPTimes, le jeu de données que nous avons construit. Nous l'avons comparé à d'autres jeux de données rassemblant les mêmes genres de documents et nous nous en sommes servi pour évaluer les performances de méthodes de production automatique de mots-clés. Nous avons aussi présenté dans les chapitre 2 et 3 les principales méthodes de production automatique de mots-clés proposées par la communauté scientifique. Pour mesurer les progrès réalisés par ces différentes méthodes et ainsi orienter les travaux de recherches il est nécessaire de comparer ces méthodes. Malheureusement, elles sont difficilement comparables. En effet, si nous examinons, par exemple, les articles de Meng et al. (2017), Florescu and Caragea (2017) et Teneva and Cheng (2017), tous publiés en 2017 à la conférence ACL, nous remarquons que ni les jeux de données, ni les paramètres, ni les protocoles d'évaluation ne sont comparables. Il est donc impossible de savoir quelle méthode est la plus performante. Il en va de même pour les méthodes en chaîne de traitement. Même si les jeux de données et métriques utilisées semblent se standardiser, le processus d'évaluation ainsi que les prétraitements ne sont pas normalisés. La mise en œuvre de ces étapes peut donc impacter les performances rapportées.

C'est pourquoi, dans ce chapitre, nous conduisons une évaluation de grande envergure sur des jeux de données variant tant sur le genre de documents que sur la taille et le type d'annotation. Nous comparons également un grand nombre de méthodes qui représentent les différentes catégories des méthodes proposées par la communauté scientifique : statistiques, fondées sur les graphes et neuronales. Pour cette évaluation, nous adoptons un cadre expérimental strict avec une chaîne de pré-traitements, une sélection des candidats et un protocole d'évaluation partagé. Ce cadre expérimental va nous permettre de répondre aux questions suivantes :

1. Quels progrès ont été réalisés en matière de production de mots-clés depuis les premières méthodes statistiques ?

2. Quel est l'impact de l'utilisation de mots-clés de référence non experts pour l'entraînement et l'évaluation des méthodes de production automatique de mots-clés ?

3. Quelles méthodes et quels jeux de données devraient être utilisés pour mieux comprendre les avantages et les inconvénients des nouvelles méthodes ?

6.1 Cadre expérimental

Dans cette section, nous détaillons successivement les jeux de données, les méthodes testées, les paramètres expérimentaux et les métriques utilisées dans cette évaluation.

6.1.1 Jeux de données

Nous utilisons 9 jeux de données qui sont représentatifs des différents genres de documents classiquement employés pour évaluer les méthodes de production automatique de mots-clés. Ces jeux de données représentent aussi différents types d'annotation : auteurs, indexeurs, lecteurs et éditeurs. Les statistiques détaillées des jeux de données sélectionnés sont présentées dans la section 4.1. Nous avons regroupé ces jeux de données en trois catégories en fonction du genre et de la taille des documents qu'ils contiennent :

— articles scientifiques : ACM, SemEval-2010, PubMed ;

— notices scientifiques : Inspec, WWW, KP20k ;

— articles journalistiques : DUC-2001, KPCrowd, KPTimes.

Nous utilisons les documents tels qu'ils ont été distribués : nous ne cherchons pas à améliorer la qualité des données en procédant, par exemple, à un nettoyage plus poussé. Les travaux de Boudin et al. (2016) montrent, en effet, qu'un nettoyage plus poussé améliore significativement les performances des méthodes. Les jeux de données KP20k et PubMed, par exemple, contiennent des noms de molécules et des formules mathématiques. Ces entités gagneraient à être segmentées en un seul mot et remplacées par un mot spécial. En effet, une mauvaise segmentation de ces entités va fausser leur étiquetage morphosyntaxique ou permettre aux modèles d'en retourner des sous-chaînes qui n'ont pas de sens en dehors de leur contexte.

6.1.2 Méthodes

Dans cette section, nous listons les méthodes de base et état de l'art choisis pour notre évaluation. Toutes ces méthodes ont été réimplémentées par nos soins. Les méthodes en chaîne de traitement, en particulier, le sont dans l'outil PKE (Boudin, 2016). Toutes les méthodes sont décrites en détail dans les sections 2.3 et 3.2. Nous rappelons donc succinctement leur fonctionnement et justifions leur choix ci-dessous.

Méthodes de base Nous avons choisi trois méthodes dites « de base », chacune associée à une caractéristique couramment utilisée dans les méthodes d'extraction de mots-clés : la position, la centralité et la fréquence. Ces méthodes de base sont non supervisées, ce qui permet leur utilisation et l'analyse de leurs performances sur tous les jeux de données. Ces trois méthodes sont :

— **FirstPhrases** qui utilise la position des mots-clés comme seul indicateur de l'importance des mots. La position en est un signal fort car les textes sont généralement écrits de manière à ce que les idées les plus importantes soient exprimées en premier (Marcu, 1997). FirstPhrases extrait, ainsi, les N premiers mots-clés candidats d'un document.

— **TextRank** (Mihalcea and Tarau, 2004) est la première méthode fondée sur les graphes et elle est souvent utilisée comme base de comparaison à d'autres méthodes. Le document est tout d'abord représenté sous forme de graphe où les mots correspondent aux nœuds et les relations de cooccurrence aux arêtes. Ensuite, l'algorithme d'ordonnancement de nœuds PageRank est exécuté pour calculer la centralité de chaque mot dans le graphe.

— **TF×IDF** (Salton and Buckley, 1988) est utilisée dans de nombreuses études comparatives (Kim et al., 2010; Meng et al., 2017, *inter alia*). Ce schéma de pondération mesure la spécificité et la fréquence de chaque mot dans un document par rapport à un corpus. Cette méthode, bien que simple, obtient dans certains cas des performances compétitives avec les modèles neuronaux de bout-en-bout.

Méthodes non supervisées Nous avons ensuite sélectionné trois méthodes non supervisées de l'état de l'art qui pallient certains manques des méthodes de base :

— **PositionRank** (Florescu and Caragea, 2017) est une méthode fondée sur les graphes qui incorpore la position et la fréquence des mots dans le calcul de leur centralité. Elle combine ainsi les trois caractéristiques utilisées par les méthodes de base.

— **MPRank** (Boudin, 2018) s'appuie sur un graphe multipartite pour regrouper les mots-clés candidats sur la base de leur similarité textuelle. Cela permet de produire des mots-clés moins redondants et de mutualiser l'information des mots-clés similaires.

— **EmbedRank** (Bennani-Smires et al., 2018) pondère les mots-clés candidats par leur similarité au document. Cette similarité est établie par la distance cosinus entre les représentations denses des mots-clés candidats et du document. Les représentations denses sont obtenues grâce à une technique de plongement de phrases (Pagliardini et al., 2018) qui capture la sémantique de ces entités.

Méthodes supervisées Nous avons retenu trois méthodes supervisées état de l'art :

— **Kea** (Witten et al., 1999) est l'une des premières méthodes supervisées de production de mots-clés. Elle utilise la position et le TF×IDF des mots-clés candidats pour les classifier comme mot-clé ou non. Elle nous permet de mesurer l'écart de performance avec les méthodes neuronales récentes, bien plus complexes et gourmandes en données d'entraînement.

— **CopyRNN** (Meng et al., 2017) est une méthode neuronale fondée sur le paradigme encodeur-décodeur avec mécanisme d'attention et de copie. Le mécanisme de copie

permet au modèle de générer des mots peu fréquents qui ne font pas partie du vocabulaire de sortie mais apparaissent dans le document. Le mécanisme d'attention, quant à lui, permet au modèle de générer des mots-clés qui capturent un aspect spécifique en portant attention aux parties du document liées à cet aspect. Contrairement aux autres méthodes choisies, CopyRNN possède la capacité de produire des mots-clés absents du document grâce à son processus génératif.

— **CorrRNN** (Chen et al., 2018) étend le modèle CopyRNN dans le but de produire des mots-clés moins redondants. Pour cela, le modèle apprend à générer les mots-clés en fonction des précédents. Un mécanisme de revue permet au modèle de porter attention à ces mots-clés déjà générés et un mécanisme de couverture informe le modèle des parties du document auxquelles il a déjà porté attention.

Notons que seules les méthodes génératives de bout-en-bout ont la capacité de générer des mots-clés absents, ce qui leur confère un avantage sur les autres méthodes.

6.1.3 Paramètres expérimentaux

Nous présentons ici les paramètres expérimentaux choisis pour notre évaluation à large couverture. Nous décrivons ainsi les pré-traitements que nous appliquons aux documents, la méthode de sélection des mots-clés candidats retenue et les paramètres des méthodes.

Pré-traitement Nous pré-traitons tous les documents des jeux de données à l'aide de Stanford CoreNLP (Manning et al., 2014) pour la tokenisation, le découpage des phrases et l'étiquetage morphosyntaxique.

Sélection des candidats Cette étape est particulièrement importante pour les méthodes en chaîne de traitement. Nous devons adopter une sélection qui minimise le silence et le bruit. Pour une comparaison équitable, nous utilisons la même heuristique de sélection des candidats pour chaque méthode. Nous utilisons les séquences de noms adjacents précédés d'un ou plusieurs adjectifs décrits par l'expression régulière suivante : `A*N+`. Nous suivons la recommandation de Wang et al. (2014) en conservant seulement les séquences de moins de cinq mots. Les candidats sont ensuite filtrés en éliminant ceux qui ont moins de 3 caractères (« km », « μm »), qui contiennent des symboles non alphanumériques (« α phényl », « vs. ») ou bien des mots vides (« théorie chimique de », « analyse des flux médiatiques d' informations »).

Paramètres Nous avons implémenté les méthodes neuronales avec PyTorch (Paszke et al., 2019) grâce à la bibliothèque AllenNLP (Gardner et al., 2018). Pour les méthodes CopyRNN et CorrRNN nous utilisons les paramètres recommandés par les auteurs originaux : GRU bidirectionnel, vocabulaire de 50 000 mots et pour l'inférence, grâce à l'algorithme de recherche en faisceau, nous générons 200 mots-clés de six mots maximum.

Les méthodes en chaîne de traitement sont implémentées dans la bibliothèque `pke` (Boudin, 2016), et nous utilisons là encore les paramètres recommandés par les auteurs originaux. Pour EmbedRank, nous utilisons la méthode `sent2vec` avec le modèle pré-entraîné `wiki_bigrams` pour calculer les représentations denses des mots-clés candidats et du document.

Entraînement Les méthodes requérant un entraînement sont : TF×IDF, Kea, CopyRNN et CorrRNN.

Nous entraînons TF×IDF et Kea sur les ensembles d'entraînement s'ils sont disponibles. Si un jeu de donnée ne possède pas d'ensemble d'entraînement, nous utilisons une procédure de validation croisée d'un contre tous (*leave-one-out*) sur l'ensemble de test. C'est-à-dire que la méthode est entraînée grâce à l'ensemble des documents moins un, et cela pour chaque document.

Les méthodes CopyRNN et CorrRNN nécessitent de grandes quantités de données annotées pour être entraînées. Ainsi, nous ne pouvons le faire qu'avec KP20k et KPTimes. Les mots-clés des jeux de données d'articles scientifiques et de notices scientifiques sont inférés par les modèles entraînés sur KP20k et ceux des jeux de données d'articles journalistiques le sont par les modèles entraînés sur KPTimes. Nous avons montré dans la section 4.1.5 qu'un nombre non négligeable de documents, apparaissant dans des ensembles de test, apparaissent aussi dans l'ensemble d'entraînement de KP20k. L'entraînement d'un modèle avec ce jeu de données fausserait la comparaison des performances entre les différents jeux de données, certains seraient avantagés par rapport aux autres. En particulier, 84 % des documents de l'ensemble de test d'ACM apparaissent dans l'ensemble d'entraînement de KP20k. Pour éviter ces problèmes, nous supprimons de l'ensemble d'entraînement de KP20k les documents communs aux ensembles de test et à cet ensemble d'entraînement.

6.1.4 Métriques d'évaluation

Bien qu'il n'y ait pas de consensus quant à la métrique la plus pertinente pour évaluer la qualité des mots-clés produits automatiquement, la stratégie d'évaluation la plus répandue consiste à rapporter la F@10 ou la F@5 (voir section 4.2.2). Nous choisissons donc de rapporter la F@10 car le nombre moyen de mots-clés dans les documents des jeux de données comparés est de 11,6. Nous rapportons également une seconde métrique, la MAP, pour évaluer l'ordonnancement des mots-clés ainsi que le test de Student apparié au niveau de 0,05 pour identifier les méthodes dont les performances sont significativement supérieures à celles des méthodes de base.

Nous observons que les méthodes de bout-en-bout sont évaluées séparément sur les mots-clés présents et les mots-clés absents, alors que les méthodes en chaîne de traitement sont évaluées sur l'intégralité de la référence. Nous choisissons ici d'évaluer les méthodes sur l'ensemble de la référence.

6.1.5 Reproductibilité des résultats

Malgré l'objectif de reproduire au plus près les résultats des articles originaux, des différences peuvent apparaître lors de la réimplémentation des méthodes. Ces différences peuvent être liées à plusieurs facteurs tels que : des paramètres peu détaillés dans les articles originaux, des différences dans les pré-traitements ou le processus d'évaluation. Ainsi, pour mesurer ces différences, nous comparons dans le tableau 6.1 les performances obtenues par les méthodes que nous avons réimplémentées à celles rapportées dans les articles originaux. Nous observons ainsi quelques différences, les plus importantes concernant les méthodes neuronales CopyRNN (+2) et CorrRNN (−3,1). Mais nous considérons que ces différences ne sont pas assez significatives pour impacter la validité des conclusions de notre étude.

Méthode	**Jeu de données**	**Métrique**	**Orig.**	**Réimp.**	**Diff.**
PositionRank	WWW	F@8	12,3	11,7	-0,6
MPRank	SemEval-2010	F@10	14,5	14,3	-0,2
EmbedRank	Inspec	F@10	37,1	35,6	-1,5
CopyRNN	KP20k	F@10 sur les présents	26,2	28,2	+2
CorrRNN	ACM	F@10 sur les présents	27,8	24,7	-3,1

TABLEAU 6.1 – Comparaison des scores des méthodes que nous avons réimplémentés avec les scores qui ont été rapportés dans les articles originaux.

Pour CorrRNN cette différence de performance peut être liée à deux facteurs. Le premier concerne les différences d'entraînement : l'article original de CorrRNN ajoute une partie des jeux de données SemEval-2010 et ACM à l'ensemble d'entraînement de KP20k, étape que nous n'avons pas suivie pour notre réimplémentation. Le deuxième facteur est lié à l'évaluation. En effet, dans l'article original, la méthode CorrRNN n'est pas évaluée sur KP20k, mais sur les jeux de données d'articles scientifiques NUS, SemEval-2010 et ACM. Notre modèle réimplémenté étant entraîné sur des notices scientifiques (176 mots en moyenne) et notre GPU ne pouvant traiter l'entièreté de ces documents (9 200 mots en moyenne), nous choisissons de nous comparer sur les documents d'ACM car il est possible d'en extraire le résumé (voir section 4.1). Ensuite, dans l'article original seuls les 400 premiers documents triés par ordre alphabétique d'ACM sont utilisés pour l'évaluation. Pour mesurer l'impact de ce choix, nous évaluons notre modèle : sur l'*ensemble* du jeu de données et obtenons 23,5 de F@10 ; sur les 400 premiers documents triés par *ordre alphabétique* et obtenons 24,7 de F@10 ; sur les 400 premiers documents triés par *ordre numérique* car les identifiants sont des nombres et obtenons 29,3 de F@10. Ainsi en modifiant simplement la méthode de tri des documents nous obtenons un écart de 4,6 points de F@10. Il est donc possible que la différence que nous obtenons provienne en partie de l'évaluation.

Pour CopyRNN, les performances plus élevées de notre modèle peuvent être liées à un paramètre qui n'est pas décrit dans l'article mais apparaît dans le code partagé sur

GitHub : la taille maximale des documents. En effet, lors de l'entraînement les documents sont tronqués à 256 mots mais nous n'avons pas effectué cette étape lors de notre réimplémentation. Ainsi, le modèle, ayant accès à plus d'informations, générerait de meilleurs mots-clés.

De plus, les poids des méthodes neuronales sont initialisés aléatoirement et introduisent donc une variabilité dans les performances obtenues. Pour minimiser cette variabilité nous aurions pu entraîner plusieurs fois le même modèle et moyenner les résultats de ces différents entraînement.

Pour la méthode Kea, les performances varient radicalement en fonction des implémentations : sur SemEval-2010, Meng et al. (2017) rapporte une F@10 de 2,6 tandis que Boudin (2016) rapporte un score de 19,3. L'implémentation que nous utilisons, basée sur celle de PKE, donne dans cette étude une F@10 de 19,5. Cette grande variation de performance est également visible pour les méthodes de base en général, comme TF×IDF ou TextRank, dont les paramètres sont peu décrits dans les articles qui les utilisent. En effet, ces méthodes ne sont pas au cœur du travail et ne sont évaluées qu'à titre de comparaison.

6.2 Résultats de l'évaluation

Après avoir décrit les méthodes et jeux de données choisis, ainsi que leurs paramètres et les métriques rapportés, nous présentons dans le tableau 6.2 les résultats de notre évaluation à large couverture.

Tout d'abord, nous remarquons qu'aucune méthode ne surpasse significativement les méthodes de base sur tous les jeux de données. Ceci est plutôt surprenant car nous nous attendions à ce que les méthodes neuronales soient systématiquement meilleures que les méthodes de bases.

Concernant ces méthodes de base, nous constatons que la méthode TF×IDF est très compétitive, notamment sur les documents longs, et obtient des performances proches des méthodes non-supervisées. La méthode FirstPhrases, malgré son extrême simplicité, obtient des résultats comparables à ceux de TF×IDF. Elle est meilleure sur les documents journalistiques que sur les documents scientifiques. La méthode TextRank, quant à elle, obtient de manière consistante les plus basses performances.

Dans l'ensemble, CopyRNN obtient les meilleures performances avec, dans le cas de KPTimes, des scores de MAP dépassant 50 %. Lorsque nous examinons uniquement les méthodes non supervisées, MPRank obtient les meilleurs résultats sur l'ensemble des jeux de données. De même, il n'est pas surprenant que Kea affiche de bonnes performances sur l'ensemble des jeux de données car elle combine deux caractéristiques efficaces, comme l'attestent les bons résultats des méthodes TF×IDF et FirstPhrases. En revanche, malgré l'ajout de mécanismes visant à promouvoir la diversité dans la sortie, CorrRNN est le plus souvent moins performant que CopyRNN. Cela suggère que les contraintes de corrélation ajoutées ne sont pas efficaces pour filtrer les mauvais mots-clés.

	Articles scientifiques						*Notices scientifiques*						*Articles journalistiques*					
	PubMed		**ACM**		**SemEval**		**Inspec**		**WWW**		**KP20k**		**DUC-2001**		**KPCrowd**		**KPTimes**	
Model	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP
FirstPhrases	15,4	14,7	13,6	13,5	13,8	10,5	29,3	27,9	10,2	9,8	13,5	12,6	24,6	22,3	17,1	16,5	9,2	8,4
TextRank	1,8	1,8	2,5	2,4	3,5	2,3	35,8	31,4	8,4	5,6	10,2	7,4	21,5	19,4	7,1	9,5	2,7	2,5
TF×IDF	16,7	16,9	12,1	11,4	17,7	12,7	**36,5**	**34,4**	9,3	10,1	11,6	12,3	23,3	21,6	16,9	15,8	9,6	9,4
PositionRank	4,9	4,6	5,7	4,9	6,8	4,1	34,2	32,2	11,6†	8,4	14,1†	11,2	28,6†	**28,0**†	13,4	12,7	8,5	6,6
MPRank	15,8	15,0	11,6	11,0	14,3	10,6	30,5	29,0	10,8†	10,4	13,6†	13,3†	25,6	24,9†	**18,2**	**17,0**	11,2†	10,1†
EmbedRank	3,7	3,2	2,1	2,1	2,5	2,0	35,6	32,5	10,7†	7,7	12,4	10,0	**29,5**†	27,5†	12,4	12,4	4,0	3,3
Kea	18,6†	18,6†	14,2†	13,3	19,5†	**14,7**†	34,5	33,2	11,0†	10,9†	14,0†	13,8†	26,5†	24,5†	17,3	16,7	11,0†	10,8†
CopyRNN	**24,2**†	**25,4**†	**24,4**†	**26,3**†	**20,3**†	13,8	28,2	26,4	**22,2**†	**24,9**†	**25,4**†	**28,7**†	10,5	7,2	8,4	4,2	**39,3**†	**50,9**†
CorrRNN	20,8†	19,4†	21,1†	20,5†	19,4	10,9	27,9	23,6	19,9†	20,3†	21,8†	22,7	10,5	6,5	7,8	3,2	20,5†	20,3†

TABLEAU 6.2 – Performance des modèles de production de mots-clés. Le symbole † indique une significativité au niveau 0.05 en utilisant le t-test de Student avec toute les méthodes de base.

6.2.1 Résultats généraux

Au vu de ces observations, nous pouvons maintenant répondre à la question suivante : « Quels progrès avons-nous réalisés en matière de production de mots-clé depuis les premières méthodes ? ».

Il est clair que les méthodes basées sur les réseaux de neurones constituent aujourd'hui l'état de l'art en matière de production automatique de mots-clés, avec des scores de F@10 jusqu'à trois fois supérieurs à ceux des méthodes précédentes. Cela étant dit, CopyRNN, qui est globalement la meilleure méthode, ne parvient pas à surpasser toutes les méthodes de base sur tous les ensembles de données. Cela s'explique notamment par la capacité de généralisation limitée des méthodes neuronales comme nous l'avons vu dans la section 5.2.2. Ainsi les performances de ces méthodes se dégradent lorsqu'elles sont évaluées sur des documents différents de ceux rencontrés lors de leur entraînement. Ce phénomène est amplifié lorsque l'annotation en mots-clés des documents d'entraînement est différente de celle des documents de test. Ce manque de généralisation est d'ailleurs confirmé par les performances extrêmement faibles des méthodes CopyRNN et CorrRNN entraînées sur KPTimes et évaluées sur les articles journalistiques de DUC-2001 et KPCrowd.

Malheureusement, les méthodes non supervisées ne sont pas significativement meilleures que les méthodes de base. Nous émettons deux hypothèses pour expliquer ce constat. La première est que les méthodes que nous avons étudiées n'utilisent pas de données du domaine, ce qui peut non seulement limiter leurs performances, mais aussi, comme dans le cas d'EmbedRank qui utilise des données hors du domaine (Wikipedia), nuire à leurs performances. Notre deuxième hypothèse concerne les mots-clés absents. En effet, les méthodes non supervisées ne sont pas capables d'en produire, contrairement aux modèles génératifs neuronaux, ce qui limite encore leur potentiel.

6.2.2 Impact des mots-clés non-experts

Comme indiqué dans la section 5.2, les références provenant d'annotateurs non professionnels, tels que les auteurs et les lecteurs, présentent de fortes incohérences. Nous pouvons donc nous demander : « Quel est l'impact de l'utilisation de références non expertes sur l'entraînement et l'évaluation des méthodes de production de mots-clés ? ».

Intuitivement, nous pensons que les méthodes évaluées sur des annotations non professionnelles sont susceptibles de recevoir des scores plus faibles car ces annotations rendent leur apprentissage plus difficile (c'est-à-dire qu'attribuer des mots-clés différents à des documents traitant du même sujet peut perturber le modèle), tout en augmentant le nombre de faux négatifs lors de l'évaluation. C'est ce que nous observons dans le tableau 6.2 où les meilleurs scores pour Inspec et KPTimes, qui comportent des mots-clés indexeurs, sont plus élevés (F@10 supérieure à 30) que ceux des autres ensembles de données (F@10 toujours inférieure à 30).

La quantification précise de l'impact des annotations non expertes sur les perfor-

mances n'est pas une tâche facile car elle implique un processus de double annotation par des annotateurs experts et non experts. Par chance, une partie des documents d'Inspec se trouve également dans KP20k, ce qui nous permet d'évaluer les méthodes sur une référence indexeur et une référence auteur, puis de les comparer. La comparaison des deux annotations montre que l'annotation indexeur est bien plus complète que l'annotation auteur. Elle comprend en moyenne 10,3 mots-clés par document contre 4,5 pour l'annotation auteur.

	F@10		MAP	
Méthode	*I*	*A*	*I*	*A*
FirstPhrases	26,9	13,4	27,2	13,0
TextRank	34,5	12,0	30,7	8,9
TF×IDF	35,0	14,6	**34,0**	15,6
PositionRank	33,2	15,3	32,2	12,5
MPRank	27,9	13,7	28,5	13,4
EmbedRank	**35,3**	15,1	32,7	11,9
Kea	32,9	15,4	32,7	15,4
CopyRNN	33,8	**27,9**‡	30,5	**34,0**‡
CorrRNN	28,7	25,0	24,1	26,9
Moy.	32,0	17,0	30,3	16,8

TABLEAU 6.3 – Performance des modèles évalués sur un sous-ensemble de 64 document d'Inspec pour les références indexeurs (I) et auteurs (A). Le symbole ‡ indique la significativité par rapport à toutes les autres méthodes.

Nous présentons les performances des méthodes évaluées sur la référence indexeur et sur la référence auteur dans le tableau 6.3. Nous constatons tout d'abord que les performances des méthodes en chaîne de traitement sont presque réduites de moitié lorsque les méthodes sont évaluées sur la référence auteur, ce qui suggère que les scores rapportés dans les études précédentes sont probablement sous-estimés. Ensuite, les méthodes neuronales ne montrent pas non plus leur nette supériorité face aux mots-clés indexeurs. En effet ce sont EmbedRank et TF×IDF qui obtiennent les meilleurs résultats. Sur les mots-clés auteurs, les scores de F@10 des méthodes neuronales baissent peu en comparaison avec les autres méthodes, et augmentent même pour la MAP. Ainsi les méthodes neuronales semblent beaucoup moins impactées par la qualité de la référence pour l'évaluation. Quant à l'entraînement, nous avons montré dans la section 5.2.2 que les performances des méthodes neuronales étaient plus élevées lors de l'entraînement sur une référence éditeur (professionnelle) que sur une référence auteur. Ces résultats soulignent la nécessité de disposer de plus de jeux de données annotés par des experts pour ne pas sous-estimer les performances des méthodes lors de leurs évaluations. Ils encouragent aussi à chercher de nouvelles manières d'évaluer automatiquement ces méthodes.

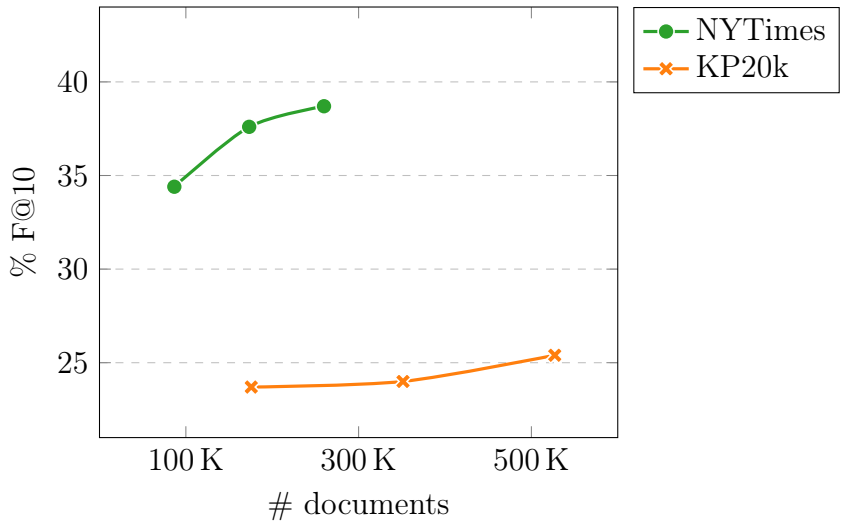


FIGURE 6.1 – Performances de CopyRNN (F@10) en fonction de la taille du jeu de données d'entraînement.

6.2.3 Courbe d'apprentissage

Les performances des méthodes de production de mots-clés neuronales dépendent fortement de la disponibilité de grandes quantités de données annotées. Pourtant, la quantité de données nécessaire pour obtenir des performances optimales n'est pas connue. Afin de répondre à cette question et pour savoir si les modèles neuronaux pourraient bénéficier de plus de données d'entraînement, nous entraînons CopyRNN en utilisant des fractions des corpus KP20k et KPTimes (33 %, 66 % et 100 %). [1] Nous présentons les performances de ces modèles dans la figure 6.1.

Nous constatons que les performances des modèles s'améliorent progressivement à mesure que la taille de l'ensemble d'entraînement grandit. L'ajout de données d'entraînement apporte des gains bien plus importants pour NYTimes $(+4,3)$ tandis que les performances de CopyRNN peinent a augmenter $(+1,7)$. La courbe des modèles entraînés avec KP20k suggère que les données disponibles ne sont actuellement pas suffisantes pour atteindre un optimum en termes de performances, tandis que celle des modèles entraînés avec NYTimes semble converger vers un optimum. En effet, l'annotation éditeur de NYTimes nécessite beaucoup moins d'exemple pour atteindre un optimum que l'annotation auteur de KP20k. Ainsi, le manque de cohérence de l'annotation dans KP20k, que nous avons souligné dans les sections 5.2 et 2.1.1, peut être la raison de ce résultat ; l'apprentissage d'un modèle est intuitivement plus difficile avec des exemples étiquetés de manière peu cohérente. D'autres travaux étudient l'augmentation de la performance des méthodes en fonction de la quantité de données disponibles. Par exemple, les travaux de Ye and Wang (2018) se placent dans un cadre où la quantité de données annotées est limitée (40 000 documents). Ils montrent qu'enrichir l'ensemble d'entraînement avec des documents an-

1. Respectivement les premiers 175 696 ; 351 393 et 527 090 échantillons pour KP20k et les premiers 86 641 ; 173 282 et 259 923 échantillons pour KPTimes.

notés automatiquement permet d'obtenir des performances comparables aux méthodes entraînées sur de plus grandes quantités de données annotées. Nous pouvons aussi mentionner les travaux de Martinc et al. (2021) qui tirent parti du transfert de connaissance d'un modèle de langue pré-entraîné. Ils affinent ce modèle pour la tâche d'identification de mots-clés avec seulement 20 000 documents annotés. Cette méthode obtient des performances comparables avec les méthodes génératives de bout-en-bout entraînées avec de plus grandes quantités de données annotées.

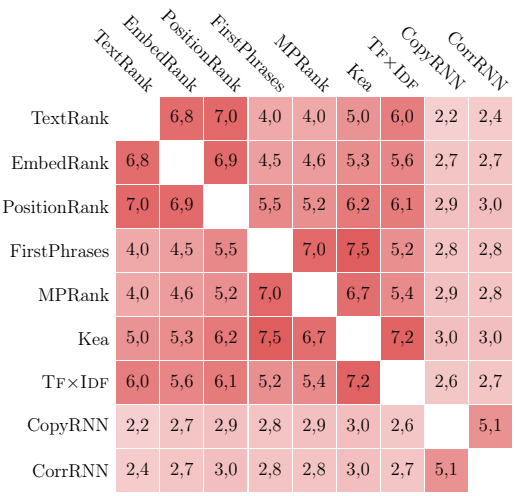
6.2.4 Choix du cadre expérimental pour l'évaluation

La troisième question à laquelle nous voulons répondre dans cette étude est la suivante : « Quelles méthodes et quels jeux de données de référence devraient être inclus dans les travaux futurs pour mieux comprendre les avantages et les inconvénients des nouvelles méthodes proposées ? »

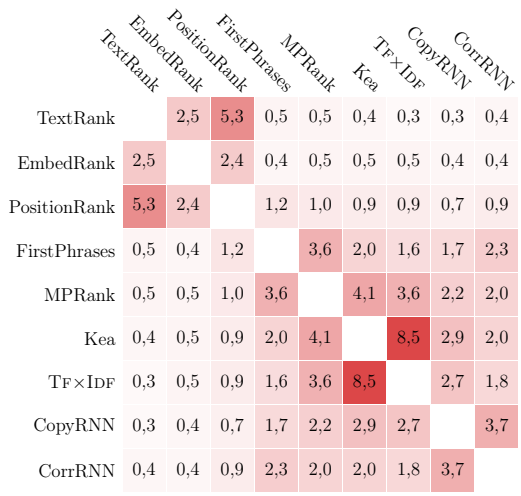
Il est important de disposer de méthodes de base dont les performances sont proches de l'état de l'art pour pouvoir s'y comparer. Les résultats de notre évaluation nous permettent de donner des indications quant au choix des méthodes les plus pertinentes à utiliser comme méthodes de base. Tout d'abord, les méthodes neuronales, lorsqu'elles sont correctement entraînées, surpassent nettement toutes les autres méthodes et représentent l'état de l'art. Ainsi et parce qu'elle obtient les meilleurs résultats, la méthode CopyRNN devrait être incluse dans les travaux futurs à des fins de comparaison. Toutefois, dans un cadre non supervisé ou dans un scénario où les données sont rares, et donc où les modèles neuronaux ne peuvent être utilisés, la situation est moins claire.

Par ailleurs, les méthodes FirstPhrases, TF×IDF, MPRank et Kea qui obtiennent des scores similaires devraient aussi être utilisés à titre de comparaison. Par contre, la méthode TextRank, bien qu'elle soit une méthode pionnière, n'est pas une bonne méthode de base car ses performances sont bien trop basses par rapport à l'état de l'art.

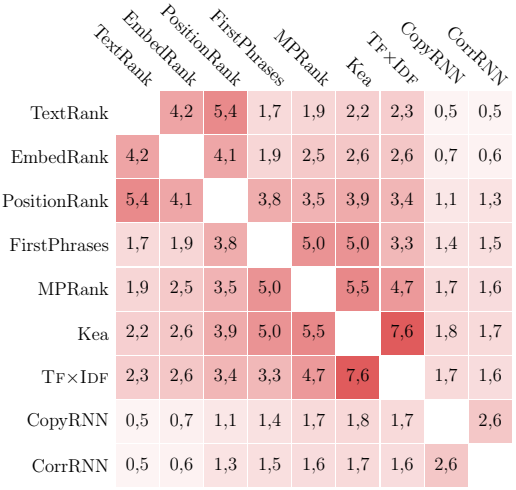
Pour éclaircir le choix des méthodes en chaîne de traitement qui devraient être incluses dans les futurs travaux, nous avons mené une série d'expériences visant à comparer les sorties des méthodes deux à deux. La motivation derrière ces expériences réside dans le fait qu'inclure plusieurs méthodes qui se comportent de manière similaire est d'un intérêt limité. Les similarités entre les sorties des méthodes, en termes de nombre de mots-clés en commun, sont représentées sous forme de *heatmap* dans la figure 6.2. Globalement, nous observons des tendances différentes pour chaque genre de documents. Plus le document est court, plus les sorties sont similaires, ce qui est principalement dû à un plus petit nombre de mots-clé candidats. Nous apercevons aussi trois clusters de méthodes dont les mots-clés produits sont similaires : TextRank, EmbedRank et PositionRank ; FirstPhrases, MPRank, Kea et TF×IDF ; et enfin les méthodes génératives CopyRNN et CorrRNN. En particulier, TF×IDF et Kea produisent des mots-clés toujours assez similaires, ce qui s'explique par l'utilisation du score de TF×IDF par Kea pour classifier les mots-clés. Notons aussi que les trois meilleures méthodes non supervisées, à savoir FirstPhrases,



(a) Notices scientifiques.



(b) Articles scientifiques.



(c) Articles journalistiques.

FIGURE 6.2 – Nombre moyen de mots-clés en commun dans les 10 premiers mots-clés de chacune des méthodes.

TF×IDF et MPRank, génèrent des mots-clés très similaires. Compte tenu de cela, et de leurs performances rapportées dans le tableau 6.2, nous soutenons que TF×IDF (ou Kea si des données d'entraînement sont disponibles) et FirstPhrases devraient être considérées comme des méthodes de base non supervisées dans les travaux futurs. Elles produisent, en effet, des mots-clés différents et sont parmi les méthodes les plus performantes.

Ces recommandations de méthodes de base ont également une incidence sur le choix des jeux de données de référence à utiliser. Les modèles neuronaux étant gourmands en données, KP20k et KPTimes sont les options par défaut pour les notices scientifiques et les articles journalistiques. Pour les articles scientifiques, nous recommandons d'utiliser SemEval pour deux raisons : 1) il est largement utilisé par les études existantes ; et 2) il fournit une référence doublement annotée (annotation auteur et lecteur) qui atténue dans une certaine mesure les incohérences d'annotation.

6.3 Conclusion

Dans ce chapitre, nous avons présenté une évaluation à grande échelle de méthodes de production automatique de mots-clés réalisée sur plusieurs jeux de données contenant des documents de genres différents. Notre objectif était de répondre à trois questions concernant les progrès réalisés par les méthodes, l'importance de la qualité de l'annotation et les méthodes et jeux de données à utiliser pour les travaux futurs. Pour atteindre cet objectif, nous avons mis en place un cadre expérimental partagé, disponible sur un dépôt public [2] afin de faciliter la reproductibilité des expériences ainsi que la réutilisation du cadre expérimental.

Nos résultats montrent que les méthodes neuronales génératives sont globalement meilleures mais que les méthodes de base (comme TF×IDF) restent compétitives. Nous avons aussi étudié l'impact des types de mots-clés de référence sur l'évaluation automatique. Ainsi, nous avons trouvé que l'évaluation sur les mots-clés auteurs sous-estime les performances des méthodes évaluées par rapport à une évaluation sur les mots-clés indexeurs, plus qualitatifs. Ces annotations indexeurs sont coûteuses en temps et en argent comme nous l'avons vu dans la section 2.1.1. C'est pourquoi, plutôt que de construire de nouvelles collections de test annotées en mots-clés indexeurs, nous préconisons le développement de nouveaux processus d'évaluation automatique.

Nous avons montré que les méthodes génératives neuronales, qui sont les plus performantes des méthodes évaluées ici, nécessitent de grandes quantités de données pour être entraînées. Pour savoir si les données actuellement disponibles sont suffisantes à leur entraînement, nous avons évalué ces méthodes après les avoir entraînées avec différentes quantités de données. Notre expérience montre que les performances des modèles augmentent avec l'ajout de données d'entraînement mais que cette augmentation est bien plus élevée avec KPTimes qu'avec KP20k. Nous en concluons que la faible cohérence

2. https://github.com/ygorg/JCDL_2020_KPE_Eval

de l'annotation auteur complexifie l'entraînement (comme nous l'avons vu dans la section 5.2.2) et limite la modélisation de cette annotation par le modèle.

Notre but avec cette étude était de recommander les méthodes à utiliser à minima pour valider les nouvelles contributions. Nous recommandons donc TF×IDF (ou Kea si des données d'apprentissage sont disponibles) et FirstPhrases comme méthodes de base ainsi que CopyRNN. Les deux premières sont non supervisées et peu gourmandes en termes de ressources, elles peuvent donc être appliquées à tout jeux de données. La dernière, CopyRNN, obtient aujourd'hui des performances état de l'art et permet de produire des mots-clés absents mais nécessite une grande puissance de calcul et est peu généralisable à de nouveaux domaines. Pour les jeux de données, nous recommandons l'utilisation de KP20k et KPTimes de par leur taille ainsi que SemEval-2010 qui contient une double annotation auteur et lecteur. Ces trois jeux de données permettent de couvrir l'ensemble des genres de documents couramment utilisés : les notices scientifiques, les articles scientifiques et les articles journalistiques.

Finalement, nos expériences ont mis en évidence plusieurs problèmes lors de l'évaluation des méthodes de production de mots-clés, notamment le choix des jeux de données à utiliser, ainsi que la qualité de leurs annotations. Dès lors, une autre façon d'évaluer l'efficacité de ces méthodes serait d'explorer leur impact dans des cadres applicatifs. Dans le chapitre suivant nous nous intéresserons à l'évaluation de la production automatique de mots-clés dans le cadre de la recherche d'information.

Chapitre 7

Impact des mots-clés en recherche d'information

Dans le chapitre 6, nous avons effectué une évaluation de 9 méthodes de production automatique de mots-clés sur autant de jeux de données dans le but de montrer l'évolution des performances des modèles proposés par la communauté scientifique. Les méthodes de bout-en-bout ont marqué un tournant dans les performances de la tâche de production automatique de mots-clés avec une amélioration significative dans le cadre d'une évaluation intrinsèque. Mais, nous avons aussi montré dans le chapitre 4 que la majorité des mots-clés de référence auxquels nous avons accès, et donc sur lesquels nous nous évaluons, sont des mots-clés auteurs. Puis dans le chapitre 5, nous avons montré que la qualité de ce type d'annotation rendait difficile l'apprentissage de modèles supervisés et résultait en une évaluation pessimiste. Il est donc important d'envisager d'autres méthodes d'évaluation pour évaluer l'utilité des mots-clés dans une tâche applicative.

Dans ce chapitre, nous allons étudier l'impact de l'enrichissement de l'indexation des documents par des mots-clés dans une tâche spécifique de recherche d'information : la recherche d'articles scientifiques. Notre objectif est de déterminer si les méthodes de production de mots-clés de l'état de l'art sont suffisamment performantes pour améliorer l'efficacité des systèmes de recherche d'information. Et dans une plus large mesure si l'enrichissement de documents par des mots-clés améliore l'efficacité des systèmes de recherche d'information.

Nous décrivons dans un premier temps le cadre expérimental dans lequel se situent nos expériences. Ensuite, nous présentons nos résultats sur l'enrichissement par des mots-clés en nous intéressant à l'impact des mots-clés produits automatiquement par rapport à ceux de référence. Nous discutons également l'impact des mots-clés présents et absents, et proposons une nouvelle catégorisation des mots-clés absents, plus pertinente dans le cadre de la recherche d'information.

7.1 Cadre expérimental

La suite de cette section détaille successivement les collections de test, les systèmes de recherche d'information, les paramètres expérimentaux et les mesures d'évaluation utilisés

dans nos expériences. Nous rappelons tout d'abord les principes généraux de la recherche d'information.

La recherche d'information (RI) est une tâche qui consiste à répondre à un besoin d'information en proposant des documents pertinents. Les besoins d'information sont exprimés sous forme de requête (un texte décrivant le besoin) qui peut être plus ou moins longue. Des documents sont rassemblés en collections, dans lesquelles chaque document est indexé – représenté de manière standardisée – pour être recherché. Pour effectuer une recherche, les documents indexés sont ordonnés par un système de recherche d'information par rapport à la requête.

7.1.1 Collections de test

Nous utilisons deux collections de test pour la tâche de recherche d'articles scientifiques. Ces collections sont composées de notices scientifiques et d'un ensemble de requêtes et jugements de pertinence correspondants. Les jugements de pertinence indiquent les documents pertinents par rapport à une requête, et donc les documents à retourner en priorité par un système de RI. La table 7.1 présente les statistiques de ces collections.

Collection	**Lang.**	**#Doc.**	**#Dmots**	**#Req.**	**#Rmots**	**#pert.**	**#mc**	**%abs**
NTCIR-2	en	322 058	156,8	49	11,3	28,8	4,8	38,1
ACM-CR	en	102 510	158,6	169	80,0	2,9	3,1	46,4

TABLEAU 7.1 – Statistiques des collections de test NTCIR-2 et ACM-CR. La table présente le nombre de documents (#Doc.) des collections et leur nombre moyens de mots (#Dmots) ; le nombre de requête (#Req.), leur nombre moyen de mots (#Rmots) et le nombre moyen de document pertinent par requêtes (#pert.) ; le nombre moyen de mots-clés (#mc) par document et le ratio de mots-clés absent (%abs) par document.

NTCIR-2 (Kando, 2001) La collection de test NTCIR-2 a été constituée à l'occasion de la compétition éponyme [1] pour une tâche de recherche d'articles scientifiques ad-hoc. Elle contient 322 058 notices scientifiques en anglais et 49 requêtes avec jugements de pertinence. La plupart (98,6 %) des documents contiennent des mots-clés auteurs (4,8 par document en moyenne). Les documents couvrent de nombreux domaines, des sciences formelles aux sciences sociales et humaines – bien que la moitié des documents concernent l'ingénierie et l'informatique. Les requêtes décrivent des besoins d'informations (un exemple de requête est présenté dans la figure 7.1) et sont associées à un ou plusieurs domaines de recherche (le champ `<FIELD>` des requêtes). Dans nos expériences, nous utilisons les requêtes courtes (le champ `<DESCRIPTION>` des requêtes) avec les jugements de pertinence binaires (c'est-à-dire qu'un document est soit « pertinent » soit « non pertinent »). Les requêtes comprennent en plus de la description courte et du domaine : un titre court (`<TITLE>`), une

1. https://research.nii.ac.jp/ntcir/

```
<TOPIC q=0145>
<TITLE>Library location</TITLE>
<DESCRIPTION>
  Papers that discuss how the locations of public libraries affect their use
</DESCRIPTION>
<NARRATIVE>
  I want papers that discuss how the locations of public libraries affect the number
  of visitors, the number of items circulated, the sphere of use, and so on. Both
  theoretical studies and case studies satisfy this retrieval request. Papers about
  other types of libraries partially satisfy the request.
</NARRATIVE>
<CONCEPT>
  a. public library, state library, county library, municipal library,
  b. location characteristics, geographical features,
  c. numbers of visitors, statistics about visitors,
  d. volume of circulation, statistics of circulation,
  e. sphere of use
</CONCEPT>
<FIELD>
  3. Architecture, civil engineering and landscape gardening,
  8. Cultural and social science
</FIELD>
</TOPIC>
```

FIGURE 7.1 – Exemple de requête extraite du corpus NTCIR-2.

description longue du besoin d'information (`<NARRATIVE>`) et des mots-clés avec variantes (`<CONCEPT>`).

ACM-CR (Boudin, 2021) La collection de test ACM-CR contient 102 411 notices scientifiques en anglais et 169 requêtes et jugements de pertinence. Les documents sont des notices d'articles scientifiques traitant de recherches d'information publiées dans les conférences et revues des groupements d'intérêts spécifiques IR, KDD, CHI, WEB et MOD. 69,2 % des documents sont annotés en mots-clés auteurs et en contiennent en moyenne 4,5. Les requêtes sont des paragraphes d'articles scientifiques contenant des citations, ainsi les jugements de pertinence indiquent comme pertinents les articles cités et disponibles dans l'ACM DL. Cette collection permet d'évaluer la tâche de recommandation de citation en contexte, que nous considérons ici comme une tâche de recherche de documents scientifiques avec un besoin d'information exprimé de manière différente – un exemple de requête est présenté dans la figure 7.2. Par rapport à NTCIR-2 les requêtes sont plus longues 11,3 mots contre 80 pour ACM-CR, et plus bruitées. En effet, la requête de la figure 7.2 doit retourner deux articles sur des thématiques différentes : les plongements de mots et l'algorithme TextRank.

```
<top>
<num> Number: 340120402
<title> Context-Aware Term Weighting For First Stage Passage Retrieval
<desc> Description:
Most first-stage retrieval models such as BM25 and query likelihood use term
frequencies (tf) to term importance in a document. A popular alternative to tf are
graph-based methods, e.g., TextRank [6]. A few recent work investigated using word
embeddings [5] for document term weighting, but most of them only learn a global idf
-like term weight because the word embeddings are context-independent. Our work aims
to learn tf-like term weights that are context-specific.
<narr> Narrative:
</top>
```

FIGURE 7.2 – Exemple de requête extraite du corpus ACM-CR. Les documents pertinents à cette requête sont les articles référés par `[5]` et `[6]`.

7.1.2 Systèmes de recherche d'information

Une fois les documents indexés (représentés sous forme de sac de mots), il est possible de les rechercher, c'est-à-dire de les ordonner selon leur pertinence par rapport à une requête. Pour calculer la pertinence des documents indexés par rapport à une requête, nous considérons deux systèmes de recherche d'information : Okapi-BM25 (Robertson et al., 1999) et Query Likelihood (QL) (Ponte and Croft, 1998), tous deux implémentés dans l'outil `anserini` (Yang et al., 2017). Ces deux systèmes adoptent des techniques non supervisées fondées sur des statistiques de corpus pour pondérer les termes. Ils seront donc directement affectés par l'ajout de mots-clés dans un document.

Okapi-BM25 (Robertson et al., 1999) Le schéma de pondération des termes dans Okapi-BM25 est calculé par la formule décrite dans l'équation 7.1 ci-dessous. [2] Similairement à TF×IDF, BM25 donne un poids élevé aux mots spécifiques à un document et rééquilibre la fréquence des mots en fonction de la longueur du document.

$$
\begin{aligned}
score(q,d) =& \cos\left(\text{BM25}(q), \text{BM25}(d)\right) \\
\text{BM25}(d) =& \left[\text{BM25}(d,w) | w \in \text{Voc}\right] \\
\text{BM25}(d,w) =& \text{TF}_{\text{BM25}} * ln\left(\frac{N - \text{DF}(w) + 0.5}{\text{DF}(w) + 0.5} + 1\right) \\
\text{TF}_{\text{BM25}} =& \frac{\text{TF}_d(w) * (k_1 + 1)}{\text{TF}_d(w) + k_1 * (1 - b + b * \frac{|d|}{\text{Moy}_{|d|}})}
\end{aligned}
\tag{7.1}
$$

Où $\text{BM25}(d)$ est un vecteur de taille $|\text{Voc}|$ contenant le poids de chaque mot dans le document d, $\text{BM25}(d,w)$ le poids du mot w dans le document d, N est le nombre de

2. Par rapport à l'article original : $R = r = 0$ et $k_3 = 0$.

documents dans la collection, $\text{DF}(w)$ le nombre de documents dans lesquels le mot w apparaît, $\text{TF}_d(w)$ la fréquence du mot w dans le document d, $|d|$ la longueur du document d et $\text{Moy}_{|d|}$ la longueur moyenne des documents de la collection. Les paramètres b et k_1 sont des constantes permettant de modifier l'importance de la longueur du document.

Query Likelihood (QL) (Ponte and Croft, 1998) QL est un schéma de pondération qui estime le niveau de pertinence d'un document en utilisant la probabilité que la requête ait été générée par le modèle de langage du document. Plus formellement, le score de pertinence entre une requête et un document est calculé selon l'équation 7.2.

$$\begin{aligned} score(q,d) =& P(q|d) \\ P(q|d) =& \prod_{w \in q} P(w|d)^{\text{TF}_q(w)} \\ P(w|d) =& \frac{\mu}{|d|+\mu} P_{mle}(w|d) + \frac{|d|}{|d|+\mu} P_{mle}(w|C) \\ P_{mle}(w|d) =& \frac{\text{TF}_d(w)}{\sum_t \text{TF}_d(t)} \end{aligned} \tag{7.2}$$

Où $P(q|d)$ est la probabilité de générer la requête q à partir du document d, $P(w|d)$ la probabilité de générer le mot w à partir du document d, $P_{mle}(w|d)$ la probabilité unigramme du mot w dans le document d et C l'ensemble des documents de la collection.

L'usage d'un modèle de langue doit être accompagné d'un mécanisme de lissage. En effet, si un mot de la requête n'apparaît pas dans le document, $P(w|d)$ sera nul, et $score(q|d)$ aussi. Ici, nous utilisons le lissage de Dirichlet (Zhai and Lafferty, 2017) (par défaut dans `anserini`) qui combine le modèle de langue du document $P_{mle}(w|d)$ et de la collection entière $P_{mle}(w|C)$ pondéré par un paramètre μ permettant de privilégier le modèle de langue du document plus celui-ci est long.

RM3 (Abdul-Jaleel et al., 2004) Nous utilisons en sus des deux systèmes décrits plus haut une méthode de retour de pertinence simulé, nommée RM3 pour obtenir des résultats proches de l'état de l'art (Lin, 2019; Yang et al., 2019). Le retour de pertinence est une technique permettant à l'utilisateur de préciser sa requête en indiquant les documents les plus pertinents retournés dans une première phase de recherche. Grâce à ce retour une seconde requête est construite et de nouveaux documents sont retournés. Le mécanisme de retour de pertinence simulé RM3 automatise ce processus en choisissant les M termes les plus importants parmi les N documents les plus pertinents. La requête finale est une interpolation de la requête originale et de la nouvelle requête par un paramètre λ.

Pour valider le choix des 4 systèmes de recherche choisis pour nos expériences (BM25, BM25+RM3, QL et QL+RM3), nous comparons leurs performances avec celles des 3 meilleurs systèmes ayant participé à la compétition NTCIR-2. Les scores des systèmes

Système	**MAP**	**P@10**
BM25+RM3	**35,5**	**38,9**
QL+RM3	34,4	36,1
1er (Fujita and Corporation, 2001)	31,9	37,4
BM25	31,9	37,1
2nd (Murata et al., 2001)	31,3	36,1
QL	31,2	35,1
3ème (Chen et al., 2001)	26,2	33,9

TABLEAU 7.2 – Efficacité de recherche documentaire des systèmes utilisés et des meilleurs systèmes présentés à NTCIR-2.

sont mesurés à l'aide de la MAP et de la P@10 (voir chapitre 4). Nous indexons les documents de la même manière que les systèmes participants à NTCIR-2, c'est-à-dire avec le titre, le résumé et les mots-clés auteurs. Les scores sont présentés dans le tableau 7.2. Nous constatons que les systèmes choisis pour nos expériences obtiennent de bons scores, dépassant même le meilleur système participant à NTCIR-2 avec une large avance. Le 1er système utilise RM3, il n'est donc pas surprenant que BM25+RM3 et QL+RM3 soient aussi en tête du classement. Le 2d système utilise BM25 ce qui explique la similarité de scores avec le système BM25.

7.1.3 Paramètres expérimentaux

Deux configurations de base pour l'indexation des documents sont comparées : titre et résumé (T+R) ; et titre, résumé et mots-clés de référence (T+R+M). Nous ajoutons à ces deux configurations les mots-clés produits automatiquement pour déterminer si ces derniers servent simplement de substituts aux mots-clés de référence ou s'ils les complètent. Nous utilisons toujours les 5 meilleurs mots-clés produits automatiquement pour enrichir l'indexation des documents (sauf si explicitement mentionné). Ce chiffre correspond au nombre moyen de mots-clés de référence observé sur les collections de test NTCIR-2 et ACM-CR.

Nous utilisons les valeurs par défaut, choisies et justifiées par `anserini` [3], pour paramétrer nos systèmes de RI. Pour Okapi-BM25, k_1 et b sont fixés à 0,9 et 0,4 respectivement. Avec QL, μ est fixé à 1 000 dans le lissage de Dirichlet. Pour RM3, M et N sont tous les deux fixés à 10 et λ est fixé à 0,5.

7.1.4 Mesures d'évaluation

L'ordonnancement des documents par un système de RI est évalué grâce aux jugements de pertinence qui permettent de savoir quels sont les documents pertinents pour

3. Plus de détail dans le fichier `SearchArgs.java` du dépôt `github.com/castorini/anserini`

une requête. Avec ces jugements de pertinence, il est possible de calculer des métriques similaires à celles présentées dans le chapitre 4. Nous rapportons ici la MAP, calculée sur les 1 000 premiers documents pour NTCIR-2, et le R@10 pour ACM-CR (tel que recommandé dans Färber and Jatowt (2020) pour la recommandation de citation). Ces métriques sont présentées dans la section 4.2.2. Nous utilisons aussi le test de Student apparié pour évaluer la significativité statistique de nos résultats au niveau de 0,05 (Smucker et al., 2007).

7.2 Mots-clés de référence et mots-clés prédits

Dans cette section nous présentons l'impact de l'indexation de mots-clés de référence et de mots-clés prédits sur les scores de recherche d'information. Tout d'abord nous présentons les résultats généraux, nous investiguons ensuite le nombre de mots-clés à ajouter au document puis l'impact du domaine des documents sur la qualité des mots-clés prédits.

7.2.1 Impact des mots-clés sur l'indexation

Dans cette expérience, nous indexons les documents de plusieurs manières pour étudier l'impact sur la RI de ces différentes indexations. Tout d'abord, et comme base de comparaison, nous indexons seulement le titre et le résumé des documents (T+R). Nous indexons ensuite le titre, le résumé et les mots-clés de référence (T+R+M [4]). À ces deux indexations, nous ajoutons des mots-clés prédits par la méthode à base de graphe MPRank, la méthode supervisée Kea [5] et les méthodes neuronales CopyRNN et CorrRNN (décrites plus en détails dans les chapitres 1 et 4).

Les scores des systèmes de recherche d'information qui indexent les documents et différents mots-clés sont reportés dans le tableau 7.3. Les colonnes du tableau présentent les scores de MAP obtenus avec les systèmes Bm25 et QL avec et sans retour de pertinence simulé RM3. Nous obtenons des scores de MAP entre 29,5 et 32,9 pour l'indexation T+R et entre 31,2 et 35,5 pour l'indexation T+R+M . Quel que soit le système d'indexation, nous constatons que l'ajout des mots-clés de référence apporte une amélioration des scores par rapport à une indexation ne les exploitant pas. Lorsque nous ajoutons des mots-clés prédits automatiquement, nous obtenons des scores de MAP entre 29,7 et 35,0 pour l'indexation T+R et entre 31,6 et 37,1 pour l'indexation T+R+M , soit une augmentation moyenne de +0,8 et +0,5 par rapport aux configurations sans mots-clés prédits. Notons que l'ajout de mots-clés prédits augmente moins les scores dans la configuration T+R+M que la configuration T+R , 1,0 et 0,6 pour CopyRNN par exemple. Ceci est dû au fait que certains mots-clés prédits sont redondant avec ceux des auteurs. En effet, les méthodes de production de mots-clés sont conçues pour répliquer l'annotation de

4. Nous utilisons un code couleur pour faciliter l'identification de nos deux configurations de base. Les couleurs ont été choisies de manière arbitraire.

5. Entraînée sur l'ensemble de validation de KP20k.

référence. Ces résultats montrent qu'avec un système donné, l'ajout de mots-clés prédits augmente les scores de MAP pour la majorité des configurations par rapport à nos deux configurations de base sans et avec mots-clés de référence.

Nous constatons un cas où le score baisse avec la méthode CorrRNN et le système QL+RM3 pour l'indexation $T{+}R{+}M$. La baisse de score de MAP de 0,1 peut être due au mécanisme de retour de pertinence simulé qui peut faire « dériver sémantiquement » la requête originale par les termes qui lui sont ajoutés. Nous étudions ce cas plus en détail dans la section 7.2.2.

Indexation	**BM25**	**+RM3**	**QL**	**+RM3**	Moy.
$T{+}R$	29,6	32,8	29,5	32,9	31,2
+ MPRank	29,7 0,1	33,0 0,2	29,8 0,4	33,0 0,1	31,4 0,2
+ Kea (KP20k)	30,3 0,7	33,9 1,1	30,3 0,9	33,5 0,6	32,0 0,8
+ CorrRNN	31,6†2,1	**35,0**†2,2	30,9†1,4	34,1 1,2	32,9 1,7
+ CopyRNN	31,4†1,9	34,8† 2,0	31,2†1,7	33,9 1,0	32,8 1,6
$T{+}R{+}M$	31,9	35,5	31,2	34,4	33,2
+ MPRank	32,0 0,1	35,8 0,3	31,6 0,4	34,7 0,3	33,5 0,3
+ Kea (KP20k)	32,1 0,2	36,0 0,5	31,8 0,6	34,7 0,3	33,6 0,4
+ CorrRNN	32,4 0,5	36,9† 1,4	32,0 0,8	34,3 -0,1	33,9 0,7
+ CopyRNN	32,5 0,5	**37,1**†1,6	32,2 1,1	36,0† 1,6	34,4 1,2

TABLEAU 7.3 – Scores de MAP sur la collection NTCIR-2 pour les systèmes de recherche documentaire utilisant différentes configuration d'indexation. La différence de score avec la configuration sans mot-clés prédits est affiché en gris. Les symboles † et ‡ indiquent la significativité par rapport à $T{+}R$ et $T{+}R{+}M$ respectivement.

Plus généralement, les améliorations de l'efficacité de recherche sont significatives lorsque les documents sont enrichis par l'union des mots-clés prédits et des mots-clés de référence, ce qui indique que ces deux types de mots-clés se complètent. Cela suggère aussi que les mots-clés prédits sont toujours utiles pour la recherche de documents scientifiques et qu'ils gagneraient à être utilisés même lorsque des mots-clés de référence sont fournis.

Les résultats obtenus avec CopyRNN sur cette évaluation extrinsèque contredisent ceux obtenus lors de notre évaluation intrinsèque (dans le chapitre 6) qui montrait la supériorité de CopyRNN par rapport aux autres méthodes de génération de mots-clés. Ici, CopyRNN et CorrRNN obtiennent des scores similaires : pour une indexation $T{+}R$ CopyRNN et CorrRNN obtiennent des scores de MAP de 32,1 et 32,0 en moyenne ; pour une indexation $T{+}R{+}M$ CopyRNN et CorrRNN obtiennent en moyenne 33,8 et 33,7. Nous comparons ces résultats à l'évaluation intrinsèque de la production de mots-clés sur NTCIR-2 (dans le tableau 7.4) dans laquelle CopyRNN obtient une F@5 de 23,9 et CorrRNN de 22,3. Les performances de production de mots-clés de CopyRNN sont supérieures mais l'impact sur la recherche d'information est similaire. Ces constats encouragent l'utilisation de cette tâche de recherche d'information pour évaluer de manière extrinsèque la production automatique de mots-clés.

Méthode	KP20k	NTCIR-2	KPTimes	ACM-CR
MPRank	14,7	18,1	14,6	12,5
CorrRNN	23,8	22,3	11,7	**22,8**
CopyRNN	**27,8**	**23,9**	**16,5**	22,6

TABLEAU 7.4 – F@5 des méthodes de production automatique de mots-clés.

Architecture of the DNA computer

In recent years, the use of DNA computers has been advocated. A DNA computer is a problem-solving system using DNA sequences in areas such as a solution of the Hamilton problem suggested by Adleman. Solutions for specific problems in other areas have also been suggested. This search request demands articles concerned with DNA computer-like systems that involve applications of the chemical action and reaction of DNA. Articles that discuss RNA computing are considered to be relevant. Articles that deal with suggestion of executable experiments not in fact executed are also considered to be relevant. Articles that discuss simulations are considered to be relevant.

FIGURE 7.3 – Requête 146 de la collection NTCIR-2.

Dans l'ensemble, BM25+RM3 obtient les meilleurs scores, ce qui confirme les résultats précédents sur la recherche d'articles scientifiques *ad-hoc* avec des données limitées (Lin, 2019). Nous constatons que les gains d'efficacité de l'expansion des requêtes (RM3) et de l'expansion des documents (ajout de mots-clés) se complètent, ce qui suggère qu'ils fournissent des signaux de pertinence différents mais complémentaires.

7.2.2 Dérive sémantique

La figure 7.4 présente une explication de la dérive sémantique de la requête « Architecture of the DNA computer » (voir figure 7.3) avec la méthode QL+RM3. Le besoin d'information de cette requête concerne les travaux sur l'architecture des ordinateurs à ADN ; le concept important dans cette requête est « ordinateur à ADN » mais elle dérive vers l'« architecture des ordinateurs ». Nous comparons les 10 premiers documents renvoyés par la requête initiale (qui seront utilisés pour la modifier à l'aide le retour de pertinence simulée) sur les documents indexés avec ou sans mots-clés prédits par CorrRNN mais toujours avec les mots-clés de référence. Ainsi, dans les documents indexés avec tous les mots-clés (T+R+M+CorrRNN), six documents concernent l'« architecture des ordinateurs », un concept beaucoup plus général que l'« architecture des ordinateurs à ADN », et ne sont donc pas pertinents. Tandis que dans les documents indexés sans mots-clés prédits (T+R+M), seuls quatre concernent « l'architecture des ordinateurs ». Nous constatons que l'ajout des mots-clés prédits par CorrRNN a amélioré le rangs des documents mais a surtout augmenté le nombre de documents non pertinents. Ce nombre de documents non pertinents renvoyés est exacerbé par le retour de pertinence simulé RM3

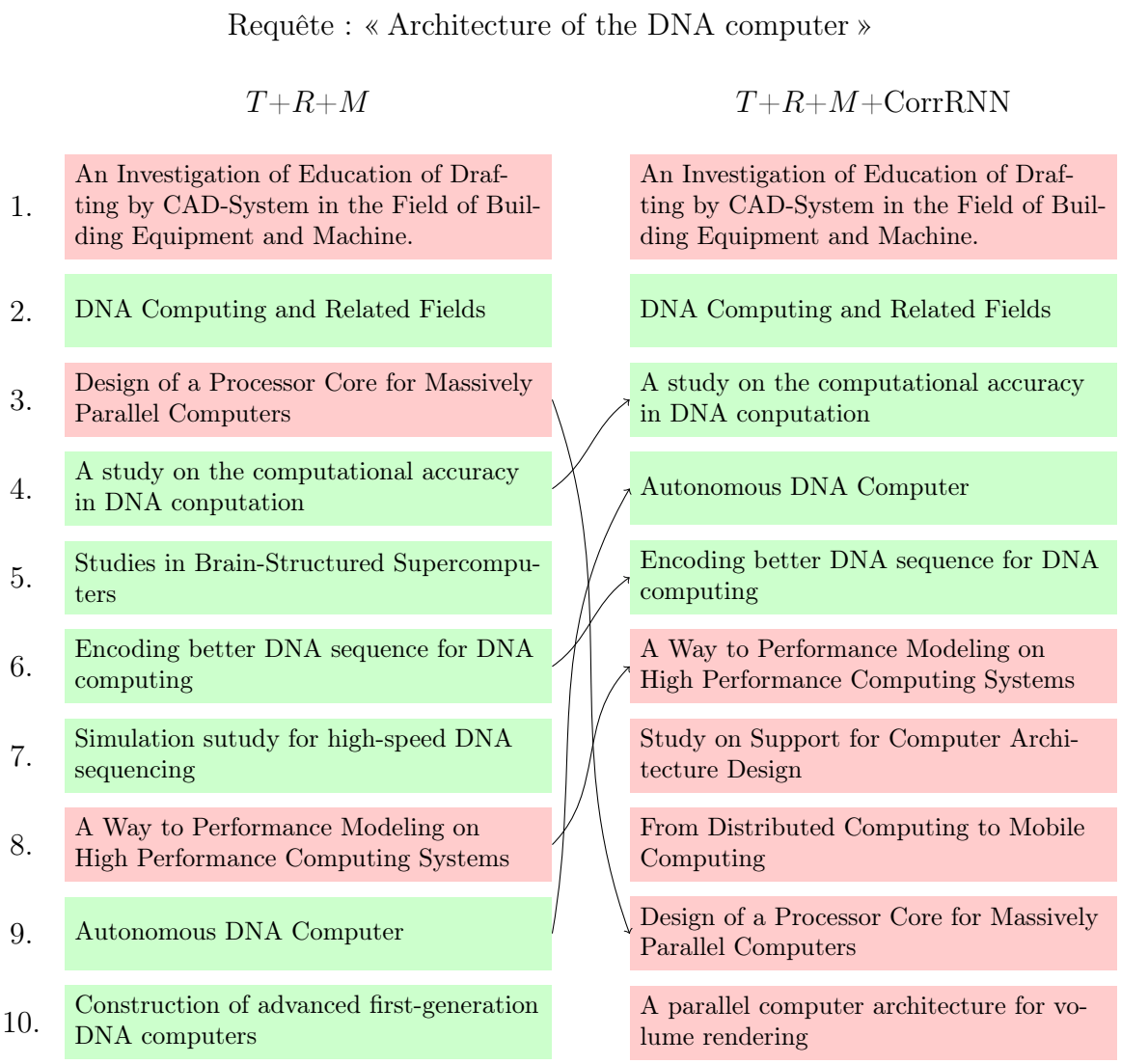


FIGURE 7.4 – Comparaison des 10 premiers documents retournés par le système QL+RM3 pour la requête 146 de la collection NTCIR-2 avec une indexation $T+R+M$ et $T+R+M+$CorrRNN. Les documents pertinents sont colorés en vert, les autres en rouge. Les flèches indiquent les documents ayant changé de rang.

qui étend la requête grâce aux mots des documents les mieux classés (avec donc plus ou moins de document non pertinents selon l'indexation). Dans la figure 7.4, les documents 2 à 5 sont pertinents et les mots-clés prédits par CorrRNN concernent l'ordinateur à ADN et le domaine de l'informatique ; les documents 1 et de 6 à 10 ne sont pas pertinents et les mots-clés prédits par CorrRNN portent uniquement sur le domaine l'informatique et non sur l'ordinateur à ADN.

7.2.3 Nombre de mots-clés automatique à ajouter

Un paramètre que nous avons délibérément laissé de côté jusqu'à présent est le nombre N de mots-clés prédits qui contrôle directement le compromis précision/rappel des méthodes de production automatique de mots-clés. Dans nos expériences de la section 7.2.1 nous avons fixé N à 5, ce qui correspond au nombre moyen de mots-clés annotés par les auteurs. Pour comprendre comment ce paramètre affecte l'efficacité de la recherche, nous avons réitéré nos expériences en faisant varier N dans l'intervalle $[0, 9]$. Les résultats de l'impact du nombre de mots-clés prédit sont présentés dans la figure 7.5.

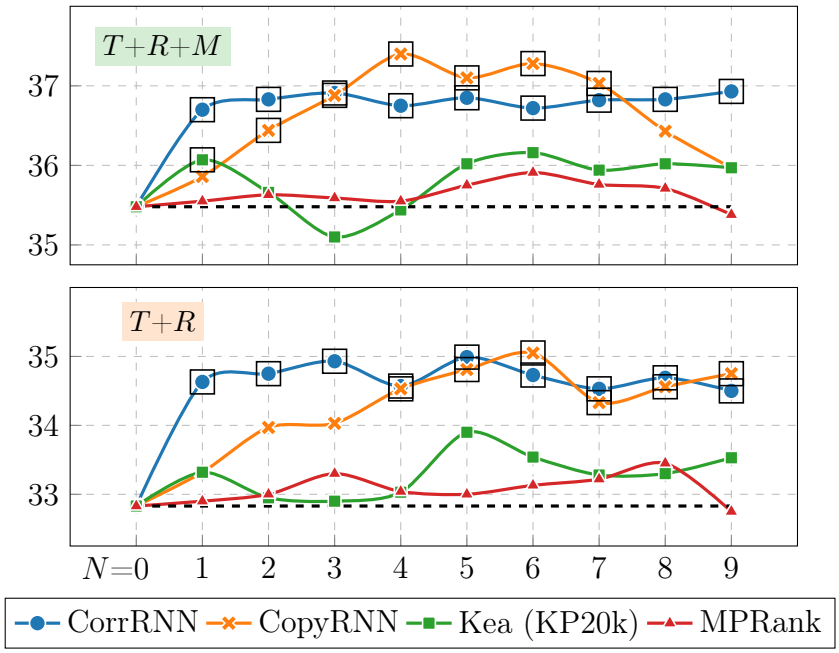


FIGURE 7.5 – Scores de MAP pour BM25+RM3 sur NTCIR-2 en fonction du nombre N de mots-clés prédits. Le symbole □ indique une amélioration significative par rapport aux résultats sans mots-clés prédits.

Deux tendances distinctes s'observent dans ces graphiques : les méthodes neuronales augmentent d'au moins un point (avec au moins deux mots-clés ajoutés) et ces augmentations sont souvent significatives ; les méthodes en chaîne de traitement augmentent d'un point au maximum et ne sont jamais significatives (sauf Kea dans la configuration $T+R+M$ avec un mot-clé ajouté). Les résultats de CorrRNN sont assez stables quel que soit le nombre de mots-clés ajoutés, le maximum étant atteint à 5 pour $T+R$ et 3 pour

$T{+}R{+}M$. L'ajout d'un nombre important de mots-clés ne dégrade pas les performances, les mots-clés sont donc cohérents avec les documents et ce dès le premier mot-clé. Les résultats de CopyRNN atteignent le maximum à 4 mots-clés ajoutés avec $T{+}R{+}M$ et 6 avec $T{+}R$. Les premiers mots-clés se complètent contrairement à CorrRNN dont seul le premier mot-clé est utile à BM25+RM3 pour sélectionner les documents pertinents. Les résultats de MPRank montrent que ses mots-clés sont peu utiles ; notons une amélioration maximale de 0,5 pour 8 mots-clés ajoutés avec $T{+}R$ et 6 avec $T{+}R{+}M$. Kea augmente significativement avec $T{+}R{+}M$ et un mot-clé ajouté. La forte augmentation avec $T{+}R$ et diminution avec $T{+}R{+}M$ sont dues à RM3 car les scores pour BM25 (non présentés ici) ne montrent pas ces fortes variations.

Les résultats de CopyRNN et CorrRNN sont proches de l'optimum pour $N = 5$, ce qui confirme empiriquement notre choix initial de 5 mots-clés établis selon le critère de production moyenne de mots-clés auteurs.

7.2.4 Impact du domaine sur les mots-clés prédits

Les méthodes neuronales de production de mots-clés présentent une capacité de généralisation limitée, ce qui signifie que leurs performances se dégradent sur des documents différents de ceux rencontrés lors de l'entraînement (Weber et al., 2018). Pour quantifier l'impact de la généralisation de ces méthodes neuronales sur l'efficacité de la recherche, nous avons divisé les requêtes de NTCIR-2 en deux ensembles disjoints : *en-domaine* pour celles qui appartiennent à des domaines de recherche présents dans KP20k, et *hors-domaine* pour les autres. Nous nous attendons donc à ce que l'ajout de mots-clés prédits par les méthodes supervisées ait un plus grand impact sur les scores des requêtes *en-domaine* que sur les requêtes *hors-domaine*. Les requêtes ont été classées en fonction de leur contenu et du champ `<FIELD>`. Les requêtes appartenant à une majorité de domaines présents dans KP20k [6] sont considérées *en-domaine* (27 requêtes) ; et celles appartenant à une majorité des domaines non présents dans KP20k [7] sont considérées *hors-domaine* (22 requêtes). Les résultats sont présentés dans le tableau 7.5.

L'examen du tableau 7.5 montre que les performances sont globalement plus faibles pour les requêtes *hors-domaine*. Les faibles performances pour l'indexation sans mots-clés prédits et pour MPRank – qui ne sont pas censés être affectés par le domaine des documents – s'expliquent par le nombre de documents pertinents pour les requêtes des deux catégories. Les requêtes *en-domaine* ont en moyenne 35 documents pertinents, tandis que les requêtes *hors-domaine* en ont 21, il est donc plus difficile de trouver les documents des requêtes *hors-domaine*. Au lieu de regarder les scores, nous nous intéressons à l'augmentation des scores de MAP (en gris) par rapport à une indexation sans mots-clés prédits (ligne « - »). Dans la configuration $T{+}R$ les mots-clés de MPRank et Kea améliorent

6. *Electricity, information and control* ; *Science* ; *Chemistry* et *Engineering*
7. *Biology and agriculture* ; *Medicine and dentistry* ; *Cultural and social science* et *Architecture, civil engineering and landscape gardening*

Méthode	T+R		T+R+M	
	E	**H**	**E**	**H**
-	33,1	32,5	36,1	34,7
MPRank	33,3 0,2	32,6 0,1	36,8 0,7	34,4 -0,3
Kea (KP20k)	34,1 1,0	33,7 1,2	37,3 1,2	34,4 -0,3
CorrRNN	34,8 1,7	**35,2**† 2,7	37,3 1,2	**36,4** 1,7
CopyRNN	**35,5**† 2,4	34,0 1,5	**37,9**† 1,8	36,1 1,4

Tableau 7.5 – Scores de MAP sur la collection NTCIR-2 pour Bm25+RM3 sur les requêtes *en-domaine* (E) et *hors-domaine* (H). La différence de score avec la configuration sans mot-clés prédits est affiché en gris. Le symbole † indique une amélioration significative par rapport à une indexation sans mots-clés produits automatiquement.

autant les scores des requêtes *en* et *hors-domaine*, ce qui conforte notre hypothèse selon laquelle les méthodes non neuronales sont peu impactées par le domaine. Dans la configuration T+R+M pour les requêtes *hors-domaine* l'ajout des mots-clés de MPRank et Kea dégradent les scores, ce qui indique que ces mots-clés ne sont pas assez similaires aux mots-clés de référence et qu'ils font baisser l'importance des mots-clés de référence. Pour CorrRNN et CopyRNN, deux méthodes supervisées neuronales, les résultats diffèrent. L'indexation utilisant les mots-clés de CopyRNN obtient les résultats attendus, l'augmentation de score est plus élevée pour les requêtes *en-domaine* que pour les requêtes *hors-domaine* (2,4 contre 1,5 pour la configuration T+R). Au contraire, l'augmentation des scores liés aux mots-clés de CorrRNN est meilleure pour les requêtes *hors-domaine* que pour les requêtes *en-domaine* (2,7 contre 1,7 pour la configuration T+R). La méthode CorrRNN semble donc mieux généraliser que CopyRNN à des documents de domaines non vus durant l'entraînement.

Méthode	E	H
MPRank	13,2	21,1
Kea (KP20k)	14,3	22,8
CorrRNN	20,5	24,6
CopyRNN	**22,0**	**27,5**

Tableau 7.6 – Performances de production de mots-clés évalués grâce à la F@5 pour les documents *en* et *hors-domaine* de NTCIR-2. Un documents est hors-domaine s'il est pertinent pour une requête hors-domaine.

Pour comparer cette évaluation extrinsèque à l'évaluation intrinsèque, nous faisons l'hypothèse que les documents pertinents pour une requête sont du même domaine que les requêtes. Ceci nous permet d'évaluer séparément les mots-clés des documents *en* et *hors-domaine*. Il y a respectivement 921 et 444 documents *en* et *hors-domaine*, et aucun de ces documents n'est pertinent à la fois pour des requêtes *en* et *hors-domaine*. Ainsi, nous présentons dans le tableau 7.6 les performances des méthodes de production de

mots-clés en fonction du domaine des documents. Étonnamment, ce tableau montre que les mots-clés produits pour les documents *hors-domaine* sont plus proches de la référence que ceux des documents *en-domaine*. Pour CopyRNN la F@5 de production de mot-clé pour les documents *en-domaine* est de 22,0 et de 27,5 pour les documents *hors-domaine*.

7.3 Mots-clés présents et absents

Les mots-clés, suivant qu'ils apparaissent ou non dans le document, jouent un rôle différent dans l'indexation. Les *mots-clés présents* mettent en évidence les parties importantes du document. Ils ajoutent de la redondance et ainsi améliorent la pondération du document. Les *mots-clés absents* sont des nouveaux termes qui étendent le contenu du document. L'attribution de mots-clés absents apparaît comme attrayante : elle pourrait atténuer la discordance de vocabulaire entre les termes de la requête et ceux des documents (Furnas et al., 1987), facilitant ainsi une meilleure récupération des documents pertinents. Cette discordance de vocabulaire est d'autant plus importante que les documents sont courts. Les bibliothèques numériques scientifiques, par exemple, indexent majoritairement les notices scientifiques, et non pas les articles entiers en raison des problèmes de licence (Huang et al., 2019) et des difficultés à les exploiter.

Study on the Structure of Index Data for Metasearch System

This paper proposes a new technique for Metasearch system, which is based on the grouping of both keywords and URLs. This technique enables metasearch systems to share information and to reflect the estimation of users' preference. With this system, users can search not only by their own keywords but by similarity of HTML documents. In this paper, we describe the principle of the grouping technique as well as the summary of the existing search systems.

Mots-clés présent : Metasearch – Search System

Mots-clés absent : Information Sharing – Information Retrieval – User's Behavior – Retrieval Support

FIGURE 7.6 – Exemple de document de la collection de test de NTCIR-2 (id : gakkai-e-0001384947).

Bien que cela ne soit pas indiqué explicitement, les travaux sur les mots-clés adoptent la définition de Meng et al. (2017) dans laquelle les mots-clés qui ne correspondent à aucune sous-séquence contiguë du texte source sont considérés comme absents. Du point de vue de la recherche d'information, où les mots pleins racinisés sont utilisés pour indexer les documents, cette définition n'est pas suffisamment explicite comme le montre l'exemple de la figure 7.6. Nous voyons que, selon cette définition, certains mots-clés absents peuvent avoir tous leurs mots dans le document source, et donc agir comme des mots-clés présents lors de l'indexation. Seule une fraction des mots qui composent ces mots-clés absents étendent véritablement le document, qui dans notre exemple est l'ensemble de mots ⌈*retrieval*, *behavior*, *support*⌋. Du point de vue de la génération de mots-clés, la définition de Meng et al. (2017) n'est pas non plus entièrement satisfaisante. En effet, les modèles génératifs

sont entraînés à générer des mots à partir d'un vocabulaire ou à copier des mots à partir du document source. L'utilisation de modèles génératifs est peut-être disproportionnée, une grande partie des mots-clés de référence sont présents dans le document et peuvent donc être copiés (voir section 6.1.1). Nous soutenons ici que la place trop importante que les modèles neuronaux donnent à la génération de mot-clés est l'une des raisons de leur faible performance. Nous allons discuter la définition actuelle des mots-clés absents et proposer une catégorisation des mots-clés absents en adéquation avec la RI qui nous permettra d'étudier plus finement l'impact des mots-clés absents et présents pour la RI.

7.3.1 Redéfinir les mots-clés absents

La définition usuelle des mots-clés présents et absents a été proposée par Meng et al. (2017) comme suit : « *nous désignons les phrases qui ne correspondent à aucune sous-séquence contiguë du texte source comme des mots-clés absents et celles qui correspondent entièrement à une partie du texte comme des mots-clés présents* ».[8] Nous soutenons que cette définition n'est pas assez précise pour deux raisons. Tout d'abord, les pré-traitements à appliquer pour faire correspondre le mot-clé et le texte ne sont pas explicités, cette étape est donc à la discrétion de l'implémentation. Ensuite, la définition de l'absence d'un mot-clé englobe ici plusieurs phénomènes distincts (occurrence totale, occurrence partielle, pas d'occurrence du mot-clé dans le document).

Pour identifier si un mot-clé est une « sous-séquence contiguë du texte source » il est évident qu'une simple correspondance de chaîne de caractères n'est pas acceptable car elle produit des faux positifs (par ex. `supervised learning` est une sous-séquence contiguë de `unsupervised learning`). Le mot-clé et le document doivent donc être mis en correspondance au niveau des mots mis en minuscules. La racinisation doit aussi être appliquée pour traiter une partie des variantes morphologiques des mots. Ce traitement est standard dans l'indexation de documents pour la RI ainsi que pour l'évaluation des méthodes de production de mots-clés.

L'« absence » de mots-clés recoupe plusieurs phénomènes que nous divisons en trois sous-catégories selon la proportion de mots présents qu'ils contiennent. La figure 7.6 présente un exemple de ces trois sous-catégories. Certains mots-clés absents ont une partie, voire la totalité, de leurs mots (dans des formes racinées) présents dans le texte tandis que d'autres ne comportent que des mots absents du texte. Nous proposons donc une catégorisation plus fine, nommée PRMN, illustrée par l'exemple de la figure 7.6 :

Présent : mot-clé dont tous les mots apparaissent de manière contiguë et dans le même ordre que dans le texte. Par exemple « Search System » et « `[...] existing search systems` . ».

Réordonné : mot-clé dont tous les mots apparaissent dans le texte mais pas de manière contiguë et/ou dans un ordre différent. Par exemple « Information Sharing » et

8. « we denote phrases that do not match any contiguous subsequence of source text as absent keyphrases, and the ones that fully match a part of the text as present keyphrases »

« [...]to share information and [...] » ou encore le mot-clé « Search System » dont les deux mots apparaissent aussi de manière non contiguë.

Mixte : mot-clé dont certains mots (mais pas tous) apparaissent dans le texte. Par exemple « Information Retrieval » et « [...] to share information and [...] », le mot « Retrieval » n'apparaît pas dans le texte.

Non-vu : mot-clé dont aucun mot n'apparaît dans le texte. Par exemple « Retrieval Support ».

Contrairement à la classification binaire (c.-à-d. présent ou absent) de Meng et al. (2017), notre schéma de catégorisation établit une distinction entre les catégories de mots-clés qui étendent le document (Mixte et Non-vu) et celles qui ne l'étendent pas (Présent et Réordonné). Notons que cette définition peut être affinée en séparant les mots-clés Réordonnés dont les mots apparaissent de manière contiguë et les autres.

L'utilisation de cette catégorisation va nous permettre d'établir la contribution de chaque catégorie à l'efficacité de la recherche d'articles scientifiques. De plus, ce schéma fournit un nouvel éclairage pour évaluer la capacité des méthodes de génération de mots-clés à produire des mots-clés absents en comparant leurs distributions PRMN à celles observées dans les annotations de référence. En d'autres termes, pour être performante, une méthode doit imiter la distribution des mots-clés absents dans l'annotation manuelle.

7.3.2 Distribution des mots-clés de référence

Nous étudions maintenant les mots-clés de référence selon la classification PRMN pour avoir une vision plus précise du nombre de mots-clés qui étendent les documents.

			mots-clés absents		
Données	**%P**	**%R**	**%M**	**%N**	**%ma**
NTCIR-2	61,9	8,1	16,5	13,5	21,4
ACM-CR	53,6	11,7	19,3	15,4	25,5
KP20k	60,2	9,5	15,4	15,0	22,3

TABLEAU 7.7 – Proportion de mots-clés Présents, Réordonnés, Mixtes et Non-vus dans les jeux de données. Nous rapportons aussi le pourcentage de mots unique absents qui proviennent des mots-clés M et N (%ma).

Le tableau 7.7 montre la proportion de mots-clés de référence attribués par les auteurs pour chaque catégorie dans nos collections de test NTCIR-2 et ACM-CR. Nous présentons aussi à titre de comparaison le jeu de données KP20k (Meng et al., 2017) utilisé pour entraîner les modèles neuronaux de génération de mots-clés.

Nous observons des distributions similaires dans les différents ensembles de données : les mots-clés absents représentent environ 40 % du nombre total de mots-clés. ACM-CR présente le ratio de mots-clés Présent le plus faible (53,6 %) et les pourcentages les plus

élevés de mots-clés Réordonnés (11,7 %) et Mixtes (19,3 %). Les distributions PRMN de NTCIR-2 et KP20k sont similaires ; KP20k comporte néanmoins 15,0 % de mots-clés Non-vu alors que NTCIR-2 n'en comporte que 13,5 %. La plupart des mots-clés absents (selon la définition de Meng) appartiennent aux catégories Mixtes et Non-vus.

Pour avoir une idée précise du nombre de nouveaux mots ajoutés lors de l'indexation des mots-clés absents, nous calculons le ratio de mots uniques absents du document (%ma) qui proviennent des mots-clés. Nous constatons que seulement 1/4 des mots uniques composants les mots-clés étendent les documents. Pourtant, comme nous le verrons plus loin, cette petite fraction de mots nouveaux est à l'origine d'une grande partie des gains observés dans l'efficacité de la recherche.

7.3.3 Impact des mots-clés de référence

Après avoir étudié la distribution des mots-clés de référence selon la catégorisation PRMN, nous étudions dans cette expérience l'impact sur la RI de ces différentes catégories. Pour cela, nous utilisons comme base de comparaison l'indexation du titre et du résumé des documents (T+R) puis indexons les différentes catégories de mots-clés PRMN. Nous comparons donc chaque catégorie individuellement mais aussi les mots-clés Présents et Réordonnés qui n'ajoutent pas de nouveaux mots au document, les mots-clés Mixtes et Non-vus qui étendent le document avec de nouveaux mots et enfin les mots-clés absents (Réordonnés, Mixtes et Non-vus).

Le tableau 7.8 présente les résultats des systèmes de recherche sur ces documents enrichis avec ces différents ensembles de mots-clés. Nous constatons que l'ajout de mots-clés améliore toujours l'efficacité de recherche pour NTCIR-2 et seulement avec Bm25 pour ACM-CR. Un examen plus poussé révèle que les gains les plus importants sont obtenus avec les mots-clés Mixtes et Non-vus : pour NTCIR-2, les scores maximaux sont atteints grâce aux mots-clés Mixtes (30,8 pour Bm25 et 33,9 pour Bm25+RM3) ; pour ACM-CR, ils sont atteints avec les mots-clés Non-vus (36,2 et 33,8).

Les mots-clés qui étendent le document (ligne « Exp. ») sont bien plus impactants que les mots-clés qui améliorent la pondération (ligne « Pond. »). Les scores d'efficacité de recherche en fonction du nombre de mots-clés ajoutés montrent que l'ajout de 1,5 mots-clés M et N augmente plus que l'ajout de 3,3 mots-clés Présents et Réordonnés pour Bm25 et Bm25+RM3. Ce schéma d'augmentation se retrouve pour ACM-CR avec Bm25. Cette observation, et le fait que le nombre de mots-clés qui étendent le document (ligne « Exp. ») est comparativement faible (moins d'un en moyenne), démontre que l'expansion des documents est plus efficace que la mise en avant des mots importants du document. Cette conclusion est confirmée par les scores plus élevés en ajoutant les mots-clés Mixtes et Non-vus (34,3) qu'en ajoutant les mots-clés Présents et Réordonnés (33,8).

De manière surprenante, pour ACM-CR, la combinaison de l'expansion de requête (+RM3) et l'ajout de mots-clés entraînent une baisse des résultats. Cette baisse peut s'expliquer par le bruit que contiennent les requêtes. En effet, les requêtes de cette collec-

	⌈ NTCIR-2 (MAP) ⌉			⌈ ACM-CR (R@10) ⌉		
index	BM25	+RM3	**#mc**	BM25	+RM3	**#mc**
T+R	29,6	32,8	-	35,6	34,1	-
+ Présent	30,7† 1,2	33,5 0,6	2,9	36,0 0,4	34,1 -0,0	1,6
+ Réordonné	29,8 0,2	33,5 0,6	0,4	35,4 -0,2	33,4 -0,7	0,3
+ Mixte	**30,8**† 1,2	**33,9** 1,0	0,8	**36,2** 0,6	33,4 -0,7	0,6
+ Non-vu	29,7 0,1	**33,9** 1,1	0,7	**36,2** 0,6	**33,8** -0,3	0,5
Pond. (P+R)	30,6† 1,1	33,8 1,0	3,3	35,8 0,2	32,4 -1,7	2,0
Exp. (M+N)	**30,8**† 1,3	**34,3** 1,5	1,5	**37,2** 1,6	**33,4** -0,7	1,1
Absent (R+M+N)	30,8† 1,2	34,9† 2,0	1,8	36,6 1,0	34,1 0,0	1,5
+ P+R+M+N	31,9†‡2,3	35,5†‡2,7	4,7	36,7 1,1	32,9 -1,2	3,1

TABLEAU 7.8 – Efficacité de recherche de BM25 et BM25+RM3 en utilisant différentes configurations d'indexation. Est aussi indiqué le nombre moyen de mots-clés ajoutés (#mc). Les symboles † et ‡ indiquent une amélioration significative par rapport à, respectivement, l'indexation T+R et Présent.

tion sont des paragraphes extraits d'articles scientifiques qui peuvent ne pas suffisamment circonscrire la recherche. Dans l'exemple de la figure 7.2, les documents recherchés pertinents portent sur les méthodes à base de graphe et les plongements de mots. Les requêtes expriment ces sujets mais abordent aussi d'autres thématiques comme les systèmes de recherche d'information. L'expansion de requête peut donc facilement faire dériver la requête originale vers ces autres sujets. La baisse des résultats d'ACM-CR peut aussi être expliquée par l'incomplétude des jugements de pertinence. Certains articles retournés répondent au besoin d'informations mais ne font pas partie des jugements de pertinence. En effet, de par la création automatique de cette collection, les jugements de pertinence incluent seulement les articles cités dans le document mais n'incluent pas les documents pertinents qui ne sont pas cités.

7.3.4 Impact des mots-clés générés présents et absents

Dans cette dernière expérience, nous examinons les mots-clés générés par les méthodes neuronales selon le schéma PRMN. Nous nous intéressons d'abord à la distribution de ces mots-clés selon la catégorisation PRMN puis à l'impact de ces mots-clés sur la RI.

Le tableau 7.9 montre les distributions sur les catégories PRMN pour nos deux méthodes neuronales : *CopyRNN* et *CorrRNN* (détaillées dans le chapitre 3). Nous observons que les mots-clés prédits sont en grande majorité Présents et que les méthodes neuronales ont donc du mal à produire des mots-clés comportant des mots absents du document (3,3 % pour les Mixtes et Non-vus de CorrRNN).

D'après nos expériences présentées dans la section 7.2.4 (voir tableau 7.3), l'extraction

Modèle	%P	%R	%M	%N	F@5
CorrRNN	89,7	7,1	2,5	0,8	22,1
CopyRNN	96,9	1,3	0,9	0,9	24,0

TABLEAU 7.9 – Proportion de mots-clés Présent, Réordonné, Mixte et Non-vu dans les 5 meilleurs mots-clés sur le jeu de donnée NTCIR-2. La F@5 est calculée sur les mots-clés de référence.

non supervisée de mots-clés n'améliore pas de manière significative l'efficacité de la recherche. La méthode MPRank dégrade même parfois les scores. À l'inverse, les méthodes neuronales augmentent significativement les scores. Cette augmentation peut être due à leur capacité à produire des mots-clés absents. Pour vérifier cette hypothèse, nous évaluons séparément l'impact des mots-clés présents (Présents et Réordonnés) et absents (Mixtes et Non-vus) avec les méthodes neuronales CopyRNN et CorrRNN et les configurations $T{+}R$ et $T{+}R{+}M$. Nous nous attendons à ce que l'ajout des mots-clés prédits absents augmentent plus les scores que l'ajout des mots-clés présents. Ce résultat conforterait nos résultats de la section 7.2.1 qui montraient que les mots-clés de référence Mixtes et Non-vus étaient responsables de la majorité des gains de scores.

	$T{+}R$ MAP		#mc	$T{+}R{+}M$ MAP		#mc	F@5
-	32,8		0,0	35,5		4,8	
CorrRNN	**35,0**†	2,2	5,0	**36,9**†	1,4	9,7	22,1
Pond. (P+R)	34,6†	1,8	5,0	36,7	1,2	9,7	25,5
Exp. (M+N)	33,4	0,6	1,9	35,8	0,3	5,2	1,7
CopyRNN	34,8†	2,0	5,0	37,1†	1,6	9,7	24,0
Pond. (P+R)	**35,0**†	2,2	5,0	**37,5**†	2,0	9,7	27,6
Exp. (M+N)	32,2	-0,6	3,2	34,5	-1,0	6,8	0,7

TABLEAU 7.10 – Scores de MAP sur la collection NTCIR-2 pour BM25+RM3 dans les configurations $T{+}R$ et $T{+}R{+}M$. Le symbole † indique une amélioration significative par rapport à l'indexation $T{+}R$. « Pond. » dénote l'ajout des mots-clés Présents et Réordonnés qui améliorent la pondération des documents. « Exp. » dénote l'ajout des mots-clés Mixtes et Non-vus qui étendent le document. CopyRNN et CorrRNN dénotent l'ajout des mots-clés sans distinction.

Les résultats présentés dans le tableau 7.10 ne répondent pas à nos attentes. L'ajout des mots-clés prédits présents (P+R) augmente significativement l'efficacité de RI par rapport aux indexations de base (sauf pour CorrRNN dans la configuration $T{+}R{+}M$). Mais l'ajout des mots-clés prédits absents (M+N) apporte peu d'amélioration pour CorrRNN (0,6 pour $T{+}R$) et dégrade les performances pour CopyRNN ($-0{,}6$ pour $T{+}R$). Pour CorrRNN, la combinaison des mots-clés présents et absents améliore plus les résultats ($+2{,}0$ pour $T{+}R{+}M$) que l'ajout des seuls mots-clés présents ($+1{,}6$ pour $T{+}R{+}M$).

Ainsi, les mots-clés absents complètent efficacement les mots-clés présents, ce qui n'est pas le cas pour CopyRNN. Ces résultats décevants pour les mots-clés absents s'expliquent par le fait que les modèles génératifs génèrent peu de mots-clés absents, et que ce peu de mots-clés absents correspond peu à la référence (voir la colonne F@5 du tableau 7.10).

Ces résultats montrent qu'enrichir les documents avec les seuls mots-clés absents (M+N) 0améliore peu ou pas les scores alors que des améliorations significatives sont systématiquement notées avec les mots-clés présents (P+R).

7.4 Conclusion

L'objectif de ce chapitre était d'évaluer la qualité des mots-clés produits par des méthodes automatiques dans le cadre de la recherche d'information. Nous avons pour cela étudié l'impact de l'indexation des mots-clés sur la tâche de recherche d'articles scientifiques *ad-hoc* et de recommandation de citation en contexte.

Nos résultats montrent que les mots-clés prédits complètent les mots-clés de référence : ces deux types de mots-clés améliorent individuellement et conjointement l'efficacité des tâches évaluées. L'utilisation de mots-clés produits automatiquement est donc utile même lorsque des mots-clés de référence sont déjà présents. Les mots-clés prédits par les méthodes neuronales sont les seuls à améliorer l'efficacité de recherche de manière significative. Ceux produits par les méthodes non-supervisées ne sont pas assez qualitatifs pour cela. L'étude de l'impact du domaine des documents sur les mots-clés prédits montre que les méthodes neuronales, à l'inverse des méthodes non neuronales, sont influencées par le domaine des documents. Ainsi nous trouvons que la méthode CopyRNN à une faible capacité de généralisation alors que CorrRNN obtient de meilleurs résultats en situation de généralisation. La comparaison entre l'évaluation intrinsèque et l'évaluation extrinsèque par la RI fournit des signaux différents pour plusieurs de nos expériences. Ce constat encourage le développement de différentes méthodes d'évaluation. Il met en exergue l'importance de réaliser ces deux méthodes d'évaluation.

Dans un second temps, nous avons défini une catégorisation fine des mots-clés absents (la catégorisation PRMN : Présent, Réordonné, Mixte, Non-vu), plus précise que la classification binaire couramment utilisée. Cette catégorisation nous a permis d'étudier l'impact de chacune de ces catégories de mots-clés pour les mots-clés de référence et les mots-clés prédits. Notre analyse montre que les mots-clés qui étendent le document (Mixtes et Non-vus), ne représentent que 30 % des mots-clés associés aux documents et ajoutent 1/4 de mots nouveaux aux documents. L'étude de l'impact de ces mots-clés montre que les mots-clés qui ajoutent de nouveaux mots aux documents sont à l'origine de la majorité des gains de scores sur les tâches évaluées. Malgré l'amélioration significative qu'ils produisent, les mots-clés des méthodes neuronales sont en très grande majorité présents, c-à-d. qu'ils n'ajoutent pas de nouveaux mots aux documents. La production de mots-clés absents quelle que soit la méthode (neuronale ou non) reste donc une problématique ouverte pour l'utilisation des mots-clés produits automatiquement.

Chapitre 8

Conclusion

Dans cette thèse nous nous sommes intéressés à la tâche de production automatique de mots-clés. L'objectif de cette tâche est d'associer à un document des mots-clés qui représentent ses concepts les plus importants. Ces mots-clés servent, entre autres, à indexer les documents dans les bibliothèques scientifiques numériques pour en faciliter la recherche, ou encore pour faciliter la navigation dans ces bibliothèques grâce à la recherche à facette.

Les méthodes actuelles de production automatique de mots-clés peuvent être séparées en deux catégories : les méthodes en chaîne de traitement (sélection de candidats puis leur pondération) et les méthodes de bout-en-bout (en une seule étape). Nous pouvons créer une dichotomie entre les méthodes qui extraient des mots-clés (qui apparaissent dans le document) et les méthodes qui en génèrent (qui apparaissent ou non dans le document). La mise à disposition à partir de 2017 d'un grand jeu de données (KP20k), a permis le développement de méthodes neuronales, aujourd'hui état de l'art. Ces méthodes reposent sur l'architecture encodeur-décodeur empruntée au domaine de la traduction automatique. Ces méthodes génératives ont actuellement deux défauts : les mots-clés produits sont très redondants, et très peu de mots-clés absents sont effectivement produits. Malgré cela, ces méthodes ont de très bons résultats sur les mots-clés présents.

L'évaluation des méthodes de production automatique de mots-clés consiste à comparer, de manière exacte, les mots-clés produits, à un ensemble de mots-clés de référence. Ce processus d'évaluation ne permet pas de prendre en compte les variantes des mots-clés, qu'elles soient syntaxiques ou sémantiques par exemple. Ainsi ce processus sous-estime les performances des méthodes et ne permet pas non plus de refléter l'utilité des mots-clés dans le cadre de tâches applicatives.

8.1 Contributions

Nos contributions principales portent sur la constitution d'un jeu de données d'articles journalistiques, une étude empirique sur les méthodes de production automatique de mots-clés, la proposition d'un nouveau processus d'évaluation extrinsèque pour ces méthodes et la proposition d'une catégorisation des mots-clés absents.

Le jeu de données KPTimes, notre première contribution, est une nouvelle ressource pour l'entraînement et l'évaluation de la tâche de production de mots-clés. Il est composé d'articles journalistiques annotés par des éditeurs. Sa taille permet l'entraînement des

modèles neuronaux. Les documents sont annotés de manière semi-automatique grâce à un système d'indexation contrôlée, cette annotation est ensuite vérifiée et enrichie par les éditeurs. Ce processus d'annotation est plus consistant que l'annotation par les auteurs et simplifie l'apprentissage des modèles neuronaux. KPTimes enrichit l'offre d'entraînement et d'évaluation des méthodes de production de mots-clés sur un nouveau domaine, non couvert par les ressources existantes. Nous avons ensuite appliqué les méthodes état de l'art de production de mots-clés sur KPTimes. Nous avons constaté que les méthodes extractives étaient moins performantes sur KPTimes que sur les autres jeux de données, ce qui s'explique par le nombre plus important de mots-clés absents qu'il contient. Par contre, les méthodes neuronales qui peuvent générer des mots-clés présents et absents sont plus performantes sur KPTimes que sur KP20k. Ce jeu de données nous a également permis de réaliser une première étude sur la capacité de généralisation des modèles neuronaux. Cette étude a montré la meilleure capacité de généralisation de CopyRNN lorsqu'il est entraîné sur le jeu de données scientifiques KP20k que sur le jeu de données généralistes KPTimes. Ce résultat est surprenant étant donné la meilleure qualité des mots-clés de KPTimes mais est en fait lié à la faible capacité du modèle entraîné sur KP20k à produire des mots-clés absents. En effet, il est moins risqué, pour avoir un rapport avec le document, de produire un mot-clé présent, limité au document, plutôt qu'un mot-clé absent. L'amélioration de la production de mots-clés absents semble donc une piste à explorer pour cette problématique de généralisation des modèles génératifs.

Notre seconde contribution est une évaluation à grande échelle des méthodes de production automatique de mots-clés. Notre objectif était d'évaluer les méthodes dans un cadre strict pour pouvoir comparer leur performances ainsi que de mesurer l'impact de la qualité des types d'annotation, en particulier entre une annotation auteur et une annotation indexeur. Nos résultats montrent que les récentes méthodes neuronales sont globalement meilleures que les méthodes statistiques classiques comme TF×IDF ou les méthodes non supervisées comme MultipartiteRank mais que ces dernières restent compétitives en particulier sur la ressource Inspec bénéficiant d'une indexation professionnelle. L'étude de l'impact des références auteur et indexeur sur l'évaluation automatique montre que la référence auteur sous-estime la performance de toutes les méthodes. En effet, les scores obtenus grâce à l'évaluation avec la référence auteur sont toujours moins importants qu'avec la référence indexeur.
La quantité de données nécessaire à l'entraînement de modèles neuronaux performants a été peu discutée pour la tâche de génération automatique de mots-clés. Ainsi, pour quantifier cette grandeur, nous avons évalué ces méthodes en faisant varier la taille de l'ensemble d'apprentissage. Nos résultats indiquent que l'ajout de données d'entraînement apporte une amélioration importante de la F-mesure pour une annotation en mots-clés effectuée par les éditeurs. L'ajout de données annotées par les auteurs, moins consistante que l'annotation par les éditeurs, n'apporte que très peu d'amélioration de la F-mesure. Ainsi, la disponibilité d'annotation en mots-clés qualitative est donc un enjeu important pour l'entraînement de méthodes génératives performantes.

Notre troisième contribution est la proposition d'un nouveau processus d'évaluation

extrinsèque par la recherche d'information. Nous avons défini un protocole d'évaluation exploitant la bibliothèque de recherche d'information `anserini` et une collection de test composée de notices scientifiques de plusieurs domaines. Nous évaluons l'impact des mots-clés en les ajoutant aux documents à indexer, puis en exécutant les requêtes. Nous avons identifié plusieurs configurations d'indexation : documents seuls, documents et mots-clés de référence, documents et mots-clés prédits, document et mots-clés de référence et prédits. Nos résultats montrent que les mots-clés améliorent l'efficacité de recherche d'information. L'amélioration est tangible pour les mots-clés de référence et pour les mots-clés prédits. Elle est la plus importante quand mots-clés de référence et mots-clés prédits sont utilisés conjointement, ce qui montre leur complémentarité. Ce résultat démontre l'intérêt de la tâche de production de mots-clés pour la recherche d'information, ce qui n'avait jamais été démontré à ce jour. Les mots-clés prédits par les méthodes neuronales sont les seuls à améliorer l'efficacité de recherche de manière significative. Ceux produits par les méthodes non-supervisées ne sont pas assez qualitatifs pour améliorer significativement les résultats. La comparaison entre les résultats de l'évaluation intrinsèque et de l'évaluation extrinsèque par la recherche d'information nous a fourni des signaux différents pour plusieurs de nos expériences. Ce constat montre l'importance de réaliser différentes évaluations et encourage le développement de nouvelles méthodes d'évaluation.

Notre quatrième contribution est la proposition d'une nouvelle catégorisation des mots-clés constituée de quatre catégories : Présent, Réordonné, Mixte et Non-vu. Cette catégorisation, plus précise que la catégorisation présent/absent actuellement utilisée, permet de différencier les mots-clés qui étendent le document avec de nouveaux mots (Mixte et Non-vu) de ceux dont les mots apparaissent déjà dans le document (Présents et Réordonné). Notre analyse fine des mots-clés de référence montre que ceux qui étendent le document ne composent que 30 % des mots-clés de référence mais sont à l'origine de la majorité des gains de scores sur les tâches évaluées. Ce résultat atteste de l'importance de ces mots-clés pour les tâches applicatives et donc de l'importance de pouvoir les produire de manière automatique. Malgré leur impact positif, les mots-clés qui étendent le document ne sont pas simples à produire automatiquement. Les performances des méthodes génératives sur ces mots-clés le montrent. Ainsi, la qualité de ces mots-clés produits automatiquement n'est, malheureusement, pas assez élevée et leur utilisation peut faire baisser les scores des tâches applicatives. Néanmoins, ces résultats encouragent l'amélioration des méthodes génératives et le développement de nouvelles méthodes permettant de produire ces mots-clés Mixtes et Non-vus.

8.2 Perspectives

Nos contributions ont permis de faire évoluer l'évaluation des méthodes de production automatique de mots-clés avec la définition et la mise en œuvre d'une évaluation extrinsèque reproductible sur une tâche concrète : la recherche d'information sur des articles scientifiques pour l'anglais.

Notre évaluation extrinsèque a porté sur deux méthodes neuronales : CopyRNN et CorrRNN. L'étude d'un plus grand nombre de ces méthodes aurait permis d'étendre encore la portée de notre travail. Les nouvelles méthodes TGNet (Chen et al., 2019b), ExHiRD (Chen et al., 2020) et leurs équivalents entraînés grâce à l'apprentissage par renforcement (Chan et al., 2019) devraient être intégrés dans les évaluations futures.

Nous aurions pu compléter nos évaluations intrinsèque et extrinsèque automatiques par une évaluation humaine de manière à voir si l'évaluation extrinsèque était plus proche de l'évaluation manuelle que l'évaluation automatique. Une évaluation humaine des mots-clés nécessite leur annotation manuelle. Ce processus peut être effectué de différentes manières et avec différents objectifs. Jones and Paynter (2001) ont fait réaliser une annotation humaine de six documents par des étudiant·es en évaluant la pertinence des mots-clés prédits par Kea et des mots-clés de référence sur une échelle de 0 à 10. Balkan (2017) a fait réaliser une annotation humaine de 50 documents par deux indexeurs professionnels, en comparant les mots-clés prédits par Kea++ à des mots-clés de référence appartenant à un vocabulaire contrôlé. Le but de cette annotation était d'évaluer la correspondance entre les mots-clés prédits et les mots-clés de référence avec trois catégories : adéquat, partiellement adéquat et inadéquat. Les mots-clés non adéquats ont ensuite été catégorisés plus finement comme : trop général, trop spécifique, redondant ou hors-sujet. Bougouin (2015); Barreaux et al. (2017) ont fait réaliser une annotation humaine par des indexeurs professionnels dans le but d'évaluer la pertinence des mots-clés prédits par rapport au document et la perte d'information de l'ensemble de mots-clés prédits par rapport aux mots-clés de référence.
Le protocole de ce type d'évaluation est complexe à définir et à mettre en place. En effet, faire appel à de nombreux annotateurs non professionnels nécessite des procédures de contrôle de l'annotation. À l'inverse il est complexe et coûteux de disposer de plusieurs indexeurs professionnels mais l'annotation est de qualité.

Nous avions projeté de réaliser les mêmes expériences pour le français. Pour mettre en œuvre les méthodes neuronales, il aurait fallu construire un large jeu de données en français. Nous aurions pu utiliser les bases bibliographiques Pascal et Francis [1] qui rassemblent 2 279 791 notices bibliographiques d'articles publiés entre 1972 et 2015 annotées en mots-clés indexeurs. Pour notre évaluation extrinsèque, nous aurions pu utiliser la collection de recherche d'information Amaryllis (Peters, 2002) qui regroupe 148 688 notices scientifiques extraites de ces mêmes bases bibliographiques Pascal et Francis ainsi que 25 requêtes.

Notre travail sur l'évaluation extrinsèque pourrait être étendu par l'ajout de nouvelles tâches applicatives. La multiplication des évaluations extrinsèques permettrait d'évaluer l'utilité des mots-clés, et donc d'encourager l'utilisation de mots-clés produits automatiquement pour des tâches applicatives. Les évaluations extrinsèques permettraient aussi de chercher à optimiser d'autres caractéristiques que la correspondance des mots-clés prédits à ceux de référence. En effet certaines caractéristiques telles que la consistance (un

1. https://pascal-francis.inist.fr

mot-clé représente un concept), la granularité (production de mots-clés plus ou moins généraux) et l'utilité (amélioration d'une tâche applicative) des mots-clés sont laissées pour compte dans l'évaluation intrinsèque actuelle. Par exemple, des mots-clés utilisés pour indexer des documents ne doivent qu'améliorer l'efficacité de recherche mais n'ont pas besoin d'être particulièrement consistants ; des mots-clés destinés à la recherche à facette par contre, doivent être consistants et faire sens pour faciliter leur utilisation par des utilisateurs. Les mots-clés sont utilisés dans d'autres applications comme l'extraction terminologique (Lanza and Daille, 2019) et la génération de questions (Subramanian et al., 2018). Il reste néanmoins à définir un protocole précis et reproductible associé à des données d'évaluation. Pour l'extraction terminologique, les données de TermEval2020 (Rigouts Terryn et al., 2020) pourraient être utilisées mais celles-ci restent de taille restreinte. Pour la génération de questions, la banque de test SQuAD (Rajpurkar et al., 2018) comportant 100 000 questions sur des textes anglais constitue un référentiel intéressant.

Liste des publications

Publication en conférence internationale avec actes

KPTimes : A Large-Scale Dataset for Keyphrase Generation on News Documents. (Gallina et al., 2019)
Ygor Gallina, Florian Boudin and Béatrice Daille

Abstract : Keyphrase generation is the task of predicting a set of lexical units that conveys the main content of a source text. Existing datasets for keyphrase generation are only readily available for the scholarly domain and include non-expert annotations. In this paper we present KPTimes, a large-scale dataset of news texts paired with editor-curated keyphrases. Exploring the dataset, we show how editors tag documents, and how their annotations differ from those found in existing datasets. We also train and evaluate state-of-the-art neural keyphrase generation models on KPTimes to gain insights on how well they perform on the news domain. The dataset is available online at `https://github.com/ygorg/KPTimes`.

Publié dans les actes de la conférence *Association for Computational Linguistics (*Acl*)*.

Large-Scale Evaluation of Keyphrase Extraction Models. (Gallina et al., 2020)
Ygor Gallina, Florian Boudin and Béatrice Daille

Abstract : Keyphrase extraction models are usually evaluated under different, not directly comparable, experimental setups. As a result, it remains unclear how well proposed models actually perform, and how they compare to each other. In this work, we address this issue by presenting a systematic large-scale analysis of state-of-the-art keyphrase extraction models involving multiple benchmark datasets from various sources and domains. Our main results reveal that state-of-the-art models are in fact still challenged by simple baselines on some datasets. We also present new insights about the impact of using author- or reader-assigned keyphrases as a proxy for gold standard, and give recommendations for strong baselines and reliable benchmark datasets.

Publié dans les actes de la conférence *Joint Conference of Digital Libraries (*Jcdl*)*.

Keyphrase Generation for Scientific Document Retrieval. (Boudin et al., 2020)
Florian Boudin, Ygor Gallina and Akiko Aizawa

Abstract : Sequence-to-sequence models have lead to significant progress in keyphrase generation, but it remains unknown whether they are reliable enough to be beneficial for document retrieval. This study provides empirical evidence that such models can significantly improve retrieval performance, and introduces a new extrinsic evaluation framework that allows for a better understanding of the limitations of keyphrase generation models. Using this framework, we point out and discuss the difficulties encountered with supplementing documents with -not present in text- keyphrases, and generalizing models across domains. Our code is available at https://github.com/boudinfl/ir-using-kg.

Publié dans les actes de la conférence *Association for Computational Linguistics (*Acl*)*.

Redefining Absent Keyphrases and their Effect on Retrieval Effectiveness (Boudin and Gallina, 2021)
Florian Boudin and Ygor Gallina

Abstract : Neural keyphrase generation models have recently attracted much interest due to their ability to output absent keyphrases, that is, keyphrases that do not appear in the source text. In this paper, we discuss the usefulness of absent keyphrases from an Information Retrieval (IR) perspective, and show that the commonly drawn distinction between present and absent keyphrases is not made explicit enough. We introduce a finer-grained categorization scheme that sheds more light on the impact of absent keyphrases on scientific document retrieval. Under this scheme, we find that only a fraction (around 20 %) of the words that make up keyphrases actually serves as document expansion, but that this small fraction of words is behind much of the gains observed in retrieval effectiveness. We also discuss how the proposed scheme can offer a new angle to evaluate the output of neural keyphrase generation models. Publié dans les actes de la conférence *Association for Computational Linguistics (*Acl*)*.

Publications en conférences nationales avec actes

État de l'art des méthodes d'apprentissage profond pour l'extraction automatique de termes-clés (Gallina, 2019)
Ygor Gallina

Résumé : Les termes-clés facilitent la recherche de documents dans de larges collections de données. Le coût d'annotation de document en termes-clés très élevé, c'est pourquoi les chercheurs s'intéressent à cette problématique. Dans cet article nous présentons un état de l'art sur l'extraction automatique de termes-clés en nous intéressant particulièrement aux modèles d'apprentissage profond. En effet, la récente publication d'un demi-million de documents annotés à permis le développement de modèles neuronaux profonds.

Publié dans les actes de la Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (Recital).

Bibliographie

Nasreen Abdul-Jaleel, James Allan, W. B. Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD:. Technical report, Defense Technical Information Center, Fort Belvoir, VA.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. In *The World Wide Web Conference on - WWW '19*, pages 2551–2557, San Francisco, CA, USA. ACM Press.

Muriel Amar. 1997. *Les fondements théoriques de l'indexation : une approche linguistique.* Thèse de doctorat, Université Lumière, Lyon, France.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Augenstein and Anders Søgaard. 2017. Multi-Task Learning of Keyphrase Boundary Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 341–346, Vancouver, Canada. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*.

Lorna Balkan. 2017. Automatic indexing of a variable/question bank collection using KEA. In *Systèmes d'organisation des connaissances et humanités numériques*, volume 10, pages 129–139. ISTE Group.

Sabine Barreaux, Béatrice Daille, and Évelyne Jacquey. 2017. Indexation automatique en SHS : bilan d'une expérimentation. *I2D - Information, donnees documents*, Volume 54(1) :15–17. Bibliographie_available : 0 Cairndomain : www.cairn.info Cite Par_available : 0 Publisher : A.D.B.S.

Marco Basaldella, Giorgia Chiaradia, and Carlo Tasso. 2016. Evaluating anaphora and coreference resolution to improve automatic keyphrase extraction. In *In Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, page 11, Osaka, Japan.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.

Gábor Berend. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3 :2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5 :135–146.

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

Florian Boudin. 2018. Unsupervised Keyphrase Extraction with Multipartite Graphs. In *Proceedings of NAACL-HLT 2018*. Association for Computational Linguistics.

Florian Boudin. 2021. ACM-CR: A Manually Annotated Test Collection for Citation Recommendation. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 280–281.

Florian Boudin and Ygor Gallina. 2021. Redefining Absent Keyphrases and their Effect on Retrieval Effectiveness. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 4185–4193, Online. Association for Computational Linguistics.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase Generation for Scientific Document Retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1126, Online. Association for Computational Linguistics.

Florian Boudin, Hugo Mougard, and Damien Cram. 2016. How Document Pre-processing affects Keyphrase Extraction Performance. *In Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, page 8.

Adrien Bougouin. 2015. *Indexation automatique par termes-clés en domaines de spécialité*. These de doctorat, Nantes.

Adrien Bougouin, Sabine Barreaux, Laurent Romary, Florian Boudin, and Béatrice Daille. 2016. TermITH-Eval: a French Standard-Based Resource for Keyphrase Extraction Evaluation. In *LREC - Language Resources and Evaluation Conference*, Potoroz, Slovenia.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 543–551, Nagoya, Japan.

Ronald Brandow, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5) :675–685.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509 :257–289.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1435–1446, Doha, Qatar. Association for Computational Linguistics.

Centre National RAMEAU. 2017. *Guide d'indexation RAMEAU*. Bibliothèque nationale de France, Paris.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural Keyphrase Generation via Reinforcement Learning with Adaptive Rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy. Association for Computational Linguistics.

Aitao Chen, Fredric C Gey, and Hailing Jiang. 2001. Berkeley at NTCIR-2 : Chinese, Japanese, and English IR Experiments. In *Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization*, page 9.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase Generation with Correlation Constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Wang Chen, Hou Pong Chan, Piji Li, Lidong Bing, and Irwin King. 2019a. An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2846–2856, Minneapolis, Minnesota. Association for Computational Linguistics.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive Hierarchical Decoding for Deep Keyphrase Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1105, Online. Association for Computational Linguistics.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019b. Title-Guided Encoding for Keyphrase Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01) :6268–6275.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Haskell B. Curry. 1944. The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, 2(3) :258–261.

Pascal Cuxac and Nicolas Thouvenin. 2017. Archives numériques et fouille de textes : le projet ISTEX. In *Actes de l'Atelier sur la Fouille de Textes TextMine organisé conjointement a la conference EGC Extraction et Gestion des Connaissances*, page 10, Grenoble, France.

Béatrice Daille. 2017. *Term Variation in Specialised Corpora: Characterisation, automatic discovery and applications*. John Benjamins.

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

Ehsan Doostmohammadi, Mohammad Hadi Bokaei, and Hossein Sameti. 2018. PerKey: A Persian News Corpus for Keyphrase Extraction and Generation. In *2018 9th International Symposium on Telecommunications (IST)*, pages 460–465.

Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3) :259–291.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.

Sumio Fujita and JUSTSYSTEM Corporation. 2001. Notes on the Limits of CLIR Effectiveness NTCIR-2 Evaluation Experiments at Justsystem. In *Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization*, page 8.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11) :964–971.

Michael Färber and Adam Jatowt. 2020. Citation recommendation: approaches and datasets. *International Journal on Digital Libraries*, 21(4) :375–405.

Ygor Gallina. 2019. État de l'art des méthodes d'apprentissage profond pour l'extraction automatique de termes-clés. In *21e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL)*, Toulouse, France.

Ygor Gallina, Florian Boudin, and Beatrice Daille. 2019. KPTimes: A Large-Scale Dataset for Keyphrase Generation on News Documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135, Tokyo, Japan. Association for Computational Linguistics.

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, pages 271–278, New York, NY, USA. Association for Computing Machinery.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

E. E Gbur and B. E. Trumbo. 1995. Key words and phrases : the key to scholarly visibility and efficiency in an information explosion. *Key words and phrases : the key to scholarly visibility and efficiency in an information explosion*, 49(1) :29–33. Place : Alexandria, VA Publisher : American Statistical Association.

Sujatha Das Gollapalli and Cornelia Caragea. 2014. Extracting Keyphrases from Research Papers Using Citation Networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1) :81–104.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1262–1273, Baltimore, Maryland. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8) :1735–1780.

Chien-yu Huang, Arlene Casey, Dorota Głowacka, and Alan Medlar. 2019. Holes in the Outline: Subject-dependent Abstract Quality and its Implications for Scientific Literature Search. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, CHIIR '19, pages 289–293, New York, NY, USA. Association for Computing Machinery.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, volume 10, pages 216–223, Not Known. Association for Computational Linguistics.

Anette Hulth and Beáta B. Megyesi. 2006. A Study on Automatically Extracted Keywords in Text Categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 537–544, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place : Sydney, Australia.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1) :11–21. Publisher : MCB UP Ltd.

Steve Jones and Gordon W. Paynter. 2001. Human evaluation of Kea, an automatic keyphrasing system. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, JCDL '01, pages 148–156, New York, NY, USA. Association for Computing Machinery.

N. Kando. 2001. Overview of the Second NTCIR Workshop. In *Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization.*

Nabil Khemiri and Sahbi Sidhom. 2020. From manual indexing to automatic indexing in the era of Big Data and Open Data: a state of the art. In *Multi-Conference OCTA'2019 on : Organization of Knowledge and Advanced Technologies*, volume 2 of *ISKO-Maghreb Proceedings*, pages 171–175, Tunis, Tunisia. Université de Tunis and ISKO-Maghreb Chapter. Issue : 1.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Branka Kosovac, Dana J. Vanier, and Thomas M. Froese. 2002. Use of Keyphrase Extraction Software for Creation of an AEC/FM Thesaurus. *Journal of Information Technology in Construction (ITcon)*, 5(2) :25–36.

Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large Dataset for Keyphrases Extraction. Departmental Technical Report, University of Trento.

Claudia Lanza and Béatrice Daille. 2019. Terminology systematization for Cybersecurity domain in Italian Language. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Terminologie et Intelligence Artificielle (atelier TALN-RECITAL \textbackslash& IC)*, pages 7–18, Toulouse, France. ATALA.

Jimmy Lin. 2019. The Neural Hype and Comparisons Against Weak Baselines. *ACM SIGIR Forum*, 52(2) :40–51.

Marina Litvak and Mark Last. 2008. Graph-based Keyword Extraction for Single-document Summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place : Manchester, United Kingdom.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic Keyphrase Extraction via Topic Decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3818–3824, New York, New York, USA. AAAI Press.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Daniel Marcu. 1997. The Rhetorical Parsing of Unrestricted Natural Language Texts. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–103, Madrid, Spain. Association for Computational Linguistics.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2021. TNT-KID: Transformer-based neural tagger for keyword identification. *Natural Language Engineering*, pages 1–40. Publisher : Cambridge University Press.

Luís Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P. Neto. 2012. 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'12)*, pages 399–403, Istanbul, Turkey. European Language Resources Association (ELRA).

Luís Marujo, Márcio Viveiros, and João Paulo da Silva Neto. 2011. Keyphrase Cloud Generation of Broadcast News. In *INTERSPEECH*.

Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. Human-competitive Tagging Using Automatic Keyphrase Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi. 2017. Deep keyphrase generation. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 582–592.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He. 2021. An Empirical Study on Neural Keyphrase Generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 4985–5007, Online. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank : Bringing Order into Texts. In *Proceedings of {EMNLP-04}and the 2004 Conference on Empirical Methods in Natural Language Processing*, page 8, Barcelona, Spain.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.

Marvin L. Minsky and Seymour A. Papert. 1988. *Perceptrons : expanded edition*. MIT Press, Cambridge, MA, USA.

James G. Mork, Antonio J. Jimeno Yepes, and Alan R. Aronson. 2013. The nlm medical text indexer system for indexing biomedical literature. In *Proceedings of the first Workshop on Bio-Medical Semantic Indexing and Question Answering,*, Valencia, Spain.

Josiane Mothe, Faneva Ramiandrisoa, and Michael Rasolomanana. 2018. Automatic keyphrase extraction using graph-based methods. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, pages 728–730, New York, NY, USA. Association for Computing Machinery.

Masaki Murata, Masao Utiyama, Qing Ma, Hiromi Ozaku, and Hitoshi Isahara. 2001. CRL at NTCIR2. In *Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization*, page 11.

Aurelie Neveol. 2005. *Automatisation des tâches documentaires dans un catalogue de santé en ligne.* Theses, INSA de Rouen.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase Extraction in Scientific Publications. In Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Torvik Sølvberg, and Edie Rasmussen, editors, *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, volume 4822, pages 317–326. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title : Lecture Notes in Computer Science.

Paul Over. 2001. Introduction to DUC-2001: an Intrinsic Evaluation of Generic News Text Summarization Systems.

Peng Si Ow and Thomas E. Morton. 1988. Filtered beam search in scheduling. *International Journal of Production Research*, 26(1) :35–62.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank Citation Ranking : Bringing Order to the Web. Technical report, Computer Science Department, Stanford University.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Carol Peters. 2002. *Advances in Cross-Language Information Retrieval : Third Workshop of the Cross-Language Evaluation Forum, CLEF 2002.* Springer Science & Business Media, Rome, Italy. Google-Books-ID : _Zx55edO2uQC.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA. Association for Computing Machinery.

M.F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3) :130–137. Publisher : MCB UP Ltd.

Vahed Qazvinian, Dragomir R. Radev, and Arzucan Özgür. 2010. Citation Summarization Through Keyphrase Extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 895–903, Beijing, China. Coling 2010 Organizing Committee.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. page 24.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Ayla Rigouts Terryn, Veronique Hoste, Patrick Drouin, and Els Lefever. 2020. TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. In *Proceedings of the 6th International Workshop on Computational Terminology*, pages 85–94, Marseille, France. European Language Resources Association.

S E Robertson, S Walker, and M Beaulieu. 1999. Okapi at TREC 7 : automatic ad hoc, ltering, VLC and interactive track. *Proceedings of the Seventh Text REetrieval Conference (TREC-7), 1999*, page 12.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings. *arXiv :1910.08840 [cs]*. ArXiv : 1910.08840.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5) :513–523.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12) :e26752.

T.y.s.s Santosh, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. SaSAKE: Syntax and Semantics Aware Keyphrase Extraction from Research Papers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5372–5383, Barcelona, Spain (Online). International Committee on Computational Linguistics.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11) :2673–2681.

Alexander Thorsten Schutz. 2008. *Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods*. Ph.D. thesis, NationalUniversityofIreland, Galway.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 623–632, New York, NY, USA. Association for Computing Machinery.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. Neural Models for Key Phrase Extraction and Question Generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88, Melbourne, Australia. Association for Computational Linguistics.

Si Sun, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. 2020. Joint Keyphrase Chunking and Salience Ranking with BERT. *arXiv :2004.13639 [cs]*. ArXiv : 2004.13639.

Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. DivGraphPointer: A Graph Pointer Network for Extracting Diverse Keyphrases. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 755–764, New York, NY, USA. Association for Computing Machinery.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Nedelina Teneva and Weiwei Cheng. 2017. Salience Rank: Efficient Keyphrase Extraction with Topic Modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 530–535, Vancouver, Canada. Association for Computational Linguistics.

Peter D. Turney. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2(4) :303–336.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations (ICLR)*.

Xiaojun Wan and Jianguo Xiao. 2008a. CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics - COLING '08*, volume 1, pages 969–976, Manchester, United Kingdom. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008b. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860, Chicago, Illinois. AAAI Press.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Baoxin Wang. 2018. Disconnected Recurrent Neural Networks for Text Categorization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 2311–2320, Melbourne, Australia. Association for Computational Linguistics.

Rui Wang, Wei Liu, and Chris Mcdonald. 2014. How Preprocessing Affects Unsupervised Keyphrase Extraction. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 8403*, CICLing 2014, pages 163–176, Berlin, Heidelberg. Springer-Verlag.

Noah Weber, Leena Shekhar, and Niranjan Balasubramanian. 2018. The Fine Line between Linguistic Generalization and Failure in Seq2Seq-Attention Models. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 24–27, New Orleans, Louisiana. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3) :229–256.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries - DL '99*, pages 254–255, Berkeley, California, United States. ACM Press.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open Domain Web Keyphrase Extraction Beyond Language Modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5175–5184, Hong Kong, China. Association for Computational Linguistics.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1253–1256, New York, NY, USA. Association for Computing Machinery.

Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically Examining the "Neural Hype": Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 1129–1132, New York, NY, USA. Association for Computing Machinery.

Hai Ye and Lu Wang. 2018. Semi-Supervised Learning for Neural Keyphrase Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4142–4153, Brussels, Belgium. Association for Computational Linguistics.

Jiacheng Ye, Ruijian Cai, Tao Gui, and Qi Zhang. 2021a. Heterogeneous Graph Neural Networks for Keyphrase Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2705–2715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021b. ONE2SET : Generating Diverse Keyphrases as a Set. page 11.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975, Online. Association for Computational Linguistics.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Daqing He, and Adam Trischler. 2018. Generating Diverse Numbers of Diverse Keyphrases. *arXiv :1810.05241 [cs]*. ArXiv : 1810.05241.

Torsten Zesch and Iryna Gurevych. 2009. Approximate Matching for Evaluating Keyphrase Extraction. In *Proceedings of the International Conference RANLP-2009*, pages 484–489, Borovets, Bulgaria. Association for Computational Linguistics.

Chengxiang Zhai and John Lafferty. 2017. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *ACM SIGIR Forum*, 51(2) :268–276.

Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. 2004. World Wide Web site summarization. *Web Intelligence and Agent Systems : An International Journal*, 2(1) :39–53. Publisher : IOS Press.

Jing Zhao and Yuxiang Zhang. 2019. Incorporating Linguistic Constraints into Keyphrase Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5224–5233, Florence, Italy. Association for Computational Linguistics.

Erion Çano and Ondřej Bojar. 2019. Keyphrase Generation: A Text Summarization Struggle. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 666–672, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrej Žukov Gregorič, Yoram Bachrach, and Sam Coope. 2018. Named Entity Recognition With Parallel Recurrent Neural Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 69–74, Melbourne, Australia. Association for Computational Linguistics.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

Titre : Indexation de bout-en-bout dans les bibliothèques numériques scientifiques

Mot clés : indexation automatique, mots-clés, évaluation extrinsèque, recherche d'information, génération de mots-clés, méthodes de bout en bout

Résumé : Le nombre de documents scientifiques dans les bibliothèques numériques ne cesse d'augmenter. Les mots-clés, permettant d'enrichir l'indexation de ces documents ne peuvent être annotés manuellement étant donné le volume de document à traiter. La production automatique de mots-clés est donc un enjeu important. Le cadre évaluatif le plus utilisé pour cette tâche souffre de nombreuses faiblesses qui rendent l'évaluation des nouvelles méthodes neuronales peu fiables. Notre objectif est d'identifier précisément ces faiblesses et d'y apporter des solutions selon trois axes. Dans un premier temps, nous introduisons KPTimes, un jeu de données du domaine journalistique. Il nous permet d'analyser la capacité de généralisation des méthodes neuronales. De manière surprenante, nos expériences montrent que le modèle le moins performant est celui qui généralise le mieux. Dans un deuxième temps, nous effectuons une comparaison systématique des méthodes états de l'art grâce à un cadre expérimental strict. Cette comparaison indique que les méthodes de référence comme TF×IDF sont toujours compétitives et que la qualité des mots-clés de référence a un impact fort sur la fiabilité de l'évaluation. Enfin, nous présentons un nouveau protocole d'évaluation extrinsèque basé sur la recherche d'information. Il nous permet d'évaluer l'utilité des mots-clés, une question peu abordée jusqu'à présent. Cette évaluation nous permet de mieux identifier les mots-clés importants pour la tâche de production automatique de mots-clés et d'orienter les futurs travaux.

Title: End-to-end indexation in digital scientific libraries

Keywords: automatic indexing, keywords, extrinsic evaluation, information retrieval, keyword generation, end-to-end method

Abstract: More and more scientific documents are being avaible in digital libraries. Efficient indexing is of the utmost importance for ease of access to scientific knowledge. Keywords, that supplements this indexation, can't be annotated manually given the volume of document to process. Automatic keyword production is then an important issue. The commonly used evaluation protocol has many weaknesses which make the evaluation of the recent neural models less reliable. Our goal is to precisely identify these weaknesses and to provide solutions given three axis. First, we introduce KPTimes, a dataset from the news domain. It will allow us to analyse the generalisation ability of neural models. Suprisingly, the least performant model is the most generalisable one. Then, we perform a systematic comparison of state-of-the-art methods using a strict experimental setup. This comparison shows that baselines such as TF×IDF are still competitive and that reference keywords quality have a strong impact on evaluation reliability. Finally, we introduce a new extrinsic evaluation protocol based on information retrieval. It allow us to evaluate keyphrase usefulness, an issue that has been given very little attention until now. This evaluation will help us better identify important keywords for automatic keyword production and to guide future works.