Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Classification de relation pour la génération de mots-clés absents

Maël Houbre Florian Boudin Béatrice Daille
Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France
prénom.nom@univ-nantes.fr

RÉSUMÉ

Les modèles encodeur-décodeur constituent l'état de l'art en génération de mots-clés. Cependant, malgré de nombreuses adaptations de cette architecture, générer des mots-clés absents du texte du document est toujours une tâche difficile. Cette étude montre qu'entraîner au préalable un modèle sur une tâche de classification de relation entre un document et un mot-clé, permet d'améliorer la génération de mots-clés absents.

ABSTRACT

Relation classification for absent keyphrase generation

Encoder-decoder models are the current state of the art for keyphrase generation. However, despite numerous adaptations of this architecture, generating keyphrases that are absent from the source text is still a difficult task. This study shows that training a model on predicting the relation between a document and a keyphrase improves absent keyphrase generation.

MOTS-CLÉS : Génération de mots-clés absents, classification de séquence, modèle encodeur-décodeur, indexation.

KEYWORDS: Absent keyphrase generation, sequence classification, encoder-decoder model, indexing.

1 Introduction

La génération de mots-clés consiste à générer un ensemble de mots ou expressions (le terme "*mot-clé*" est utilisé dans les deux cas) représentant les points d'intérêt d'un document. Ceux-ci sont utilisés pour différentes tâches telles que le résumé automatique (Zha, 2002; Wan *et al.*, 2007; Qazvinian *et al.*, 2010; Pasunuru & Bansal, 2018) ou l'indexation (Harter, 1975; Barker *et al.*, 1972). Ces mots-clés peuvent être présents dans le texte source du document ou absents de celui-ci. Ils sont alors appelés mots-clés présents (respectivement absents). Les mots-clés absents apportent une plus-value en enrichissant l'indexation des documents scientifiques (Boudin & Gallina, 2021). Contrairement à son équivalent extractif, la génération de mots-clés a la particularité de permettre de générer ces mots-clés absents. La génération de mots-clés a été introduite avec l'utilisation de l'architecture encodeur-décodeur (Meng *et al.*, 2017). Cependant, les mots-clés absents résultant d'une abstraction, leur génération est particulièrement difficile, et ce malgré de nombreuses adaptations du modèle encodeur-décodeur (Yuan *et al.*, 2020; Meng *et al.*, 2021; Chen *et al.*, 2020; Bahuleyan & El Asri, 2020; Ye *et al.*, 2021).

Plutôt que d'essayer d'améliorer l'architecture, plusieurs travaux se sont concentrés sur l'utilisation

Actes de l'atelier "Analyse et Recherche de Textes Scientifiques" (ARTS)@TALN 2023, pages 47–53.

de tâches support pour améliorer la représentation du texte avec des modèles pré-entraînés (Yasunaga *et al.*, 2022; Kulkarni *et al.*, 2022; Wu *et al.*, 2022). Cependant, aucun de ces travaux n'a travaillé sur la représentation de l'ensemble des mots-clés (présents et absents). Nous inspirant de ces travaux, nous introduisons une nouvelle tâche support visant à améliorer l'encodage des mots-clés et notamment des mots-clés absents.

Les contributions de cette étude sont comme suit :

- des travaux préliminaires sur une nouvelle tâche support à la génération de mots-clés : la prédiction de relation entre un mot-clé et un document.
- deux modèles pré-entraînés, utilisables avec la librairie transformers [1].

2 Méthodologie

Notre approche repose sur l'utilisation de deux affinages successifs d'un modèle pré-entraîné ; un premier affinage sur une tâche de classification suivi d'un affinage sur la génération de mots-clés. Le but de notre méthode est d'apprendre au modèle à mieux représenter la relation entre un document et ses mots-clés. En demandant au modèle de reconnaître les mots-clés auteur d'un document, nous avons pour objectif d'améliorer la représentation entre un mot-clé et son document par l'encodeur. De précédents travaux se sont appuyés sur des tâches support afin d'insister sur certains aspects importants du document tels que certains passages ciblés ou les mots-clés présents (Wu *et al.*, 2022; Kulkarni *et al.*, 2022). Cependant, aucun n'utilise l'ensemble complet des mots-clés (présents et absents) dans ces différentes tâches.

Les travaux à l'origine du modèle LinkBERT (Yasunaga *et al.*, 2022) améliorent significativement l'encodeur BERT en utilisant un graphe document-hyperlien. Ce graphe permet de récupérer des textes issus de documents liés dans le graphe pour enrichir les données d'entraînement. Le modèle est ensuite entraîné à déterminer si un passage est modifié avec du contenu provenant du document lui-même, d'un document lié ou d'un document aléatoire. Nos travaux s'inscrivent dans cette ligne en utilisant également une tâche de classification de séquence mais où le lien entre les documents est basé sur les mots-clés. Nous entraînons ensuite le modèle obtenu sur la génération de mots-clés.

Dans cet article, la tâche de classification consiste à déterminer si la séquence contient un mot-clé auteur, un mot-clé lié ou un mot-clé aléatoire. Les mots-clés auteurs d'un article sont les mot-clés attribués par les auteurs de celui-ci. Pour chaque mot-clé Y d'un document D_1, nous déterminons les documents D_k qui possèdent aussi ce mot-clé. Les mots-clés de D_k autres que Y sont appelés mots-clés liés. La figure 1 représente un exemple. Le mot-clé en bleu et rouge est un mot-clé partagé entre les documents D1 et D2. Les mots-clés bleus sont donc les mots-clés liés du document D1. Les mots-clés aléatoires (en vert sur la figure 1) sont pris aléatoirement dans l'ensemble restant des mots-clés du corpus (i.e sans les mots-clés auteur du document et ses mots-clés liés).

Pour la tâche de classification, la séquence à classifier est constituée de deux sous-séquences ; un mot-clé, puis la concaténation du titre et du résumé du document. Le mot clé est soit un mot-clé auteur du document, soit un mot-clé lié, soit un mot-clé aléatoire. Le modèle doit classifier la séquence selon trois étiquettes "auteur, lié, aléatoire". L'hypothèse est qu'à l'instar de LinkBERT, entraîner le modèle à faire cette distinction améliorera l'encodeur et permettra d'avoir une meilleure représentation entre les mots-clés et leurs documents associés.

1. https://huggingface.co/docs/transformers/index

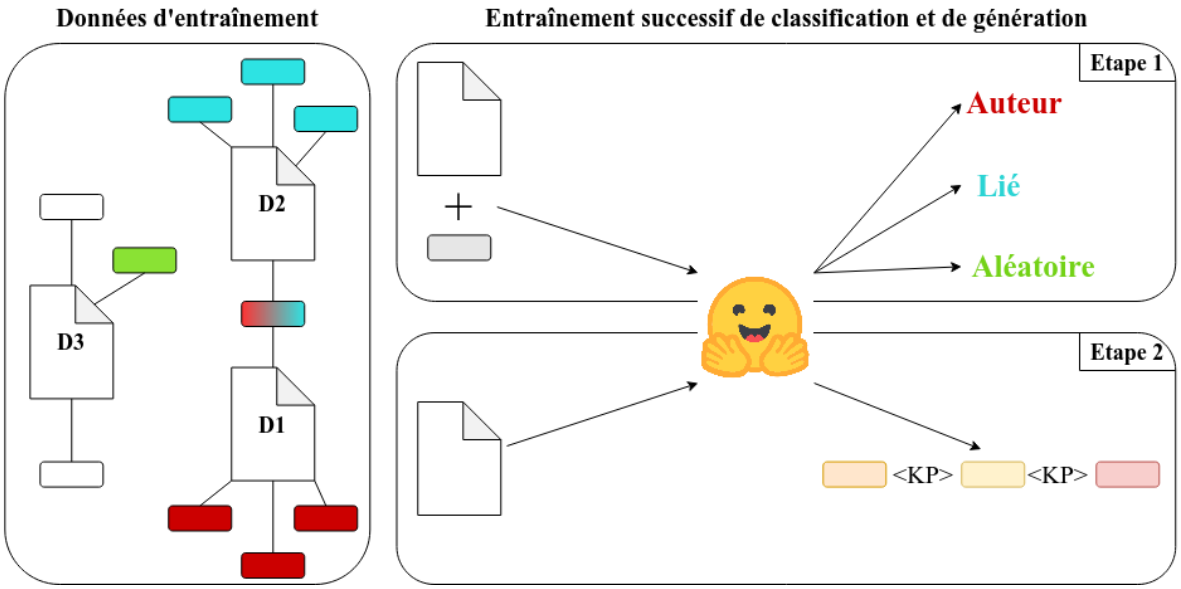

FIGURE 1 – Illustration de l'approche. Les mots-clés de D2 non communs avec D1 sont dits "liés".

Notre approche s'appuie sur le modèle BART (Lewis *et al.*, 2020). Ce modèle génératif avec une architecture encodeur-décodeur a déjà été utilisé pour la génération de mots-clés (Chowdhury *et al.*, 2022; Houbre *et al.*, 2022; Meng *et al.*, 2022) et notamment dans des travaux avec des tâches support (Kulkarni *et al.*, 2022; Wu *et al.*, 2022).

3 Expériences

Nous entraînons le modèle sur le dataset KP20k (Meng *et al.*, 2017) pour les deux tâches (classification et génération). Ce dataset contient 530 000 documents scientifiques en anglais dans le domaine des sciences informatiques issus de la bibliothèque numérique ACM Digital Library [2]. Chaque document est annoté avec en moyenne 5 mots-clés. En plus de ces mots-clés auteurs, nous ajoutons les mots-clés liés et les mots-clés aléatoires. Nous ajoutons au maximum 5 mots-clés liés et 5 mots-clés aléatoires à chaque document. Pour la classification, chaque séquence ne comporte qu'un seul mot-clé. L'ensemble d'entraînement est constitué de 8,1 millions de paires document/mot-clé. Le modèle est entraîné pendant 4 époques sur 4 cartes graphiques V100 32Go. L'entraînement de la tâche support prend 96 heures. Le modèle est ensuite entraîné sur la génération de mots-clés. Nous utilisons le paradigme One2Seq (Meng *et al.*, 2021) qui consiste pour un document en entrée, à générer l'ensemble des mots-clés dans une unique séquence. La séquence de mots-clés de référence est composée des mots-clés présents dans leur ordre d'apparition dans le texte, suivis des mots-clés absents dans l'ordre donnée par l'auteur puis des mots-clés liés et aléatoires. Le modèle est entraîné pendant 10 époques. L'entraînement de la génération de mots-clés prend 9 heures.

Conformément aux travaux à l'état de l'art, nous distinguons l'évaluation de la génération des mots-clés présents et absents. Pour les mots-clés présents, nous utilisons la F1@M et la F1@10. La F1@M (respectivement F1@10) est la f-mesure appliquée sur la première séquence de mots-clés générée par le modèle (respectivement les top 10 mots-clés générés). Pour la génération de mots-clés

2. https://dl.acm.org/

absents, nous utilisons le rappel appliqué au top 10 mots-clés générés (R@10). Pour obtenir 10 mots-clés, nous utilisons une recherche par faisceau et générons 20 séquences. Nous prenons ensuite les 10 premiers mots-clés en enlevant les répétitions. Si nous ne pouvons pas obtenir 10 mots-clés uniques, la séquence est complétée par autant d'unités "<UNK>" que nécessaire. Avant d'effectuer la comparaison, les mots-clés de la référence et les mots-clés prédits sont racinisés avec l'algorithme de Porter. La significativité statistique des résultats est vérifiée par un test de student avec $p < 0.05$.

4 Résultats et discussion

Le tableau 1 détaille les résultats des expériences sur l'ensemble de test de KP20k. Le symbole †représente une différence statistiquement significative entre les résultats des deux modèles.

Modèle	F1@M	F1@10	R@10
BART	31.4	28.5	4.7
BART classif (notre approche)	31.5	28.3	5.0†

TABLE 1 – Résultats de la génération de mots-clés présents (F1@10 et F1@M) et absents (R@10)

Les résultats du tableau 1 montrent que pour la génération de mots-clés présents, il n'y a pas de différence significative entre le modèle avec tâche support *BART classif* et celui entraîné uniquement sur la génération de mots-clés. Concernant la génération de mots-clés absents, les performances du modèle *BART classif* sont meilleures que celles du modèle BART uniquement entraîné sur la génération de mots-clés. Cependant, bien que la différence soit statistiquement significative, l'amélioration n'est que de 6.4% relatifs. La différence entre les résultats ne dépassant pas les 10%, nous qualifions plutôt cette amélioration de perceptible plutôt que de significative (Sparck Jones, 1974).

L'une des hypothèses pour lesquelles la génération de mots-clés présents n'est pas améliorée est que la tâche de classification implique un plus grand nombre de mots-clés absents que de présents. Avec 5 mots-clés liés et 5 mots-clés aléatoires supplémentaires par document, le ratio entre mots-clés présents et mots-clés absents est inversé. Affiner le choix et le nombre des mots-clés liés et aléatoires est une première voie d'amélioration. La forme de la séquence donnée en entrée de la classification peut également être une des raisons de ces faibles performances par rapport à l'état de l'art sur des architectures dédiées à la génération de mots-clés. En effet, les travaux de (Kulkarni *et al.*, 2022) et (Wu *et al.*, 2022) portaient sur le remplacement d'éléments du texte source. Ceci fournissait ainsi un contexte à l'encodeur pour distinguer le contenu "étranger" au document. Dans notre étude, le mot-clé est introduit en tout début de séquence sans contexte supplémentaire. Améliorer le paradigme pour la classification fera l'objet de prochains travaux. Une autre hypothèse est que l'utilisation de deux affinages successifs ne permet pas de tirer pleinement profit de la tâche support. L'utilisation d'un entraînement multi-tâche est une voie à explorer.

5 Conclusion

Dans cette étude, nous avons proposé une nouvelle tâche support visant à améliorer la génération de mots-clés. Nous entraînons un modèle à distinguer si un mot-clé est associé par l'auteur, extrait

de l'ensemble des mots-clés d'un document lié (i.e avec lequel le document partage un ou plusieurs mots-clés) ou aléatoire. Nous avons montré qu'un modèle BART entraîné sur cette tâche avant la génération de mots-clés, voyait ses performances significativement améliorées pour la génération de mots-clés absents. Cependant, la tâche support augmente drastiquement le temps d'entraînement pour des résultats qui ne dépassent pas l'état de l'art. De futurs travaux se concentreront sur l'amélioration de la définition de la tâche support, ainsi que son utilisation pour d'autres modèles génératifs.

6 Remerciements

Ce travail s'inscrit dans le cadre du projet ANR DELICES (ANR-19-CE38-0005) et a été effectué en utilisant les resources de calcul de GENCI-IDRIS (dossier 2022-[AD011013670]).

Références

BAHULEYAN H. & EL ASRI L. (2020). Diverse keyphrase generation with neural unlikelihood training. In *Proceedings of the 28th International Conference on Computational Linguistics*, p. 5271–5287, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.462.

BARKER F., VEAL D. & WYATT B. (1972). COMPARATIVE EFFICIENCY OF SEARCHING TITLES, ABSTRACTS, AND INDEX TERMS IN A FREE-TEXT DATA BASE. *Journal of Documentation*, **28**(1), 22–36. DOI : 10.1108/eb026527.

BENAMARA F., HATOUT N., MULLER P. & OZDOWSKA S., Éds. (2007). *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse. ATALA, IRIT.

BOUDIN F. & GALLINA Y. (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*, p. 4185–4193, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.330.

CHEN W., CHAN H. P., LI P. & KING I. (2020). Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 1095–1105, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.103.

CHOWDHURY M. F. M., ROSSIELLO G., GLASS M., MIHINDUKULASOORIYA N. & GLIOZZO A. (2022). Applying a Generic Sequence-to-Sequence Model for Simple and Effective Keyphrase Generation. *arXiv :2201.05302 [cs]*. arXiv : 2201.05302.

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

HARTER S. P. (1975). A probabilistic approach to automatic keyword indexing. Part I. On the Distribution of Specialty Words in a Technical Literature. *Journal of the American Society for Information Science*, **26**(4), 197–206. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.4630260402, DOI : 10.1002/asi.4630260402.

HOUBRE M., BOUDIN F. & DAILLE B. (2022). A large-scale dataset for biomedical keyphrase generation. In *Proceedings of the 13th International Workshop on Health Text Mining and Infor-*

mation Analysis (LOUHI), p. 47–53, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

KULKARNI M., MAHATA D., ARORA R. & BHOWMIK R. (2022). Learning Rich Representation of Keyphrases from Text. In *Findings of the Association for Computational Linguistics : NAACL 2022*, p. 891–906, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.findings-naacl.67.

LAIGNELET M. & RIOULT F. (2009). Repérer automatiquement les segments obsolescents à l'aide d'indices sémantiques et discursifs. In A. NAZARENKO & T. POIBEAU, Éds., *Actes de TALN 2009 (Traitement automatique des langues naturelles)*, Senlis : ATALA LIPN.

LANGLAIS P. & PATRY A. (2007). Enrichissement d'un lexique bilingue par analogie. In (Benamara *et al.*, 2007), p. 101–110.

LEWIS M., LIU Y., GOYAL N., GHAZVININEJAD M., MOHAMED A., LEVY O., STOYANOV V. & ZETTLEMOYER L. (2020). BART : Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7871–7880, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.703.

MENG R., WANG T., YUAN X., ZHOU Y. & HE D. (2022). General-to-Specific Transfer Labeling for Domain Adaptable Keyphrase Generation. arXiv :2208.09606 [cs].

MENG R., YUAN X., WANG T., ZHAO S., TRISCHLER A. & HE D. (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*, p. 4985–5007, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.naacl-main.396.

MENG R., ZHAO S., HAN S., HE D., BRUSILOVSKY P. & CHI Y. (2017). Deep Keyphrase Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 582–592, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1054.

PASUNURU R. & BANSAL M. (2018). Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 646–653, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-2102.

QAZVINIAN V., RADEV D. R. & ÖZGÜR A. (2010). Citation summarization through keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, p. 895–903, Beijing, China : Coling 2010 Organizing Committee.

SERETAN V. & WEHRLI E. (2007). Collocation translation based on sentence alignment and parsing. In (Benamara *et al.*, 2007), p. 401–410.

SPARCK JONES K. (1974). AUTOMATIC INDEXING. *Journal of Documentation*, **30**(4), 393–432. Publisher : MCB UP Ltd, DOI : 10.1108/eb026588.

WAN X., YANG J. & XIAO J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, p. 552–559, Prague, Czech Republic : Association for Computational Linguistics.

WU D., AHMAD W., DEV S. & CHANG K.-W. (2022). Representation learning for resource-constrained keyphrase generation. In *Findings of the Association for Computational Linguistics :*

EMNLP 2022, p. 700–716, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics.

YASUNAGA M., LESKOVEC J. & LIANG P. (2022). LinkBERT : Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 8003–8016, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.551.

YE J., GUI T., LUO Y., XU Y. & ZHANG Q. (2021). One2Set : Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 4598–4608, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.354.

YUAN X., WANG T., MENG R., THAKER K., BRUSILOVSKY P., HE D. & TRISCHLER A. (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*, p. 7961–7975, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.710.

ZHA H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, p. 113–120, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/564376.564398.