Rien ne sert de décoder, il faut chercher à point : prédiction de mots-clés inter-domaine par recherche et classement à l'aide de Sentence-Transformers fine-tunés

Saber Zahhar[*,**], Nédra Mellouli[**], Christophe Rodrigues[**], Nicolas Travers[**]

[*]Devoteam
saber.zahhar@devoteam.com
[**]De Vinci Higher Education, De Vinci Research Center, Paris, France
{prenom.nom}@devinci.fr

Résumé. La génération de mots-clés demeure encore un défi pour l'état de l'art, fortement focalisée sur les encodeurs-décodeurs neuronaux. Or, les modèles peinent à généraliser hors domaine ou même à générer des mots-clés absents satisfaisants dans leur domaine d'entraînement, qui plus est, ces modèles nécessitent d'importantes ressources de calcul et sacrifient souvent la latence pour des gains marginaux. Notre approche propose une architecture uniquement basée sur l'encodage, dédiée au classement de mots-clés issus d'un regroupement inter-domaines. En utilisant le même ensemble d'entraînement que les décodeurs comme index, chaque document de test est traité comme une requête : nous collectons les mots-clés des voisins les plus proches, puis apprenons à les classer à l'aide d'un Sentence-Transformers ajusté par fine-tuning aux différents domaines. L'apprentissage repose sur un objectif contrastif de type multiple negatives ranking, où chaque document de test est associé à un mot-clé de référence parmi un ensemble partagé de candidats négatifs. Cependant, certains mots-clés pouvant être pertinents à plusieurs documents, nous masquons leur contribution à la fonction objectif afin d'éviter de les considérer négatifs et pénaliser le modèle à tort. Cette adaptation permet de mieux modéliser les recouvrements sémantiques entre documents tout en préservant la stabilité de l'entraînement. Nous comparons notre méthode à de solides bases seq2seq, en évaluant le f-score et le rappel pour les mots-clés présents et absents, la robustesse hors domaine, la latence, les coûts d'entraînement et d'inférence ainsi que l'empreinte environnementale. Notre approche égale ou dépasse les modèles génératifs tout en réduisant fortement les limites associés à ces approches neuronales. Ce travail se positionne ainsi comme une alternative scalable aux architectures seq2seq pour l'indexation documentaire dans les bases de connaissances par mots-clés.

1 Introduction

Les bibliothèques numériques servent aujourd'hui de principal conduit pour l'accès, la préservation et l'organisation des connaissances. L'un des défis qu'elles rencontrent concerne le

processus d'indexation des publications. Traditionnellement, les bibliothèques physiques donnaient accès aux ouvrages à travers des métadonnées basiques comme l'année de publication, les noms d'auteurs ou encore une catégorisation générique (e.g. "science-fiction", "sciences sociales"). Cependant, ceci ne permet plus de répondre adéquatement aux besoins utilisateurs, notamment par leur incapacité à réellement informer sur le contenu d'un document. Les **mots-clés** sont des mots et expressions qui encapsulent les idées fondamentales d'un document. Ces mots-clés offrent une représentation complète et fidèle des travaux publiés, avec le potentiel d'améliorer la pertinence des résultats en recherche documentaire (Boudin et Gallina, 2021). Bien que l'assignement de mots-clés aux documents soit devenue une procédure indispensable à la publication, leur mise en œuvre informative et cohérente pour l'indexation reste un défi considérable. En effet, en raison des coûts associés à l'annotation manuelle professionnelle, les auteurs se voient confier cette tâche sans aucune expertise particulière (Strader, 2009). Par conséquent, les mots-clés générés sont souvent inadéquats pour la recherche interdisciplinaire (Kwon, 2018). De plus, l'état de l'art s'est focalisé sur l'extraction de mots-clés. Ces méthodes ne contribuent pas à enrichir le document, car elles ne font qu'extraire des données déjà présentes. (Boudin et Gallina, 2021) révèle que ce sont les mots-clés absents qui ont l'effet le plus prépondérant sur l'indexation documentaire grâce à leur capacité à étendre un document. Plus récemment, les chercheurs ont focalisé leur attention sur les méthodes abstractives ; cependant, les contributions restent fortement axées sur la complexification d'architectures neuronales pour un gain marginal et limité. Nous abordons ce problème à travers le paradigme du *retrieval-augmented ranking* (RAR) : à l'instar d'une génération séquentielle d'un décodeur neuronal comme proposé par le *retrieval-augmented generation* (RAG), nous exploitons un index de documents pour récupérer des candidats mots-clés que apprenons à classer avec un encodeur. Ce cadre de recherche-classement permet de capitaliser sur les annotations existantes tout en évitant le coût de la génération autorégressive. Ainsi, nous proposons [1] :

1. l'encodeur de SEARCHKEYS pour le classement de candidats inter-domaines ;
2. un modèle `BART-KP` multi-domaine servant de baseline seq2seq forte pour comparer équitablement généralisation, latence et coûts ;
3. l'ensemble des inférences mots-clés par nos modèles entraînés et ceux de l'état de l'art.
4. une évaluation multi-dimensionnelle (PRMU, inter-domaine, latence, coût et empreinte environnementale), ainsi que la mise à disposition de l'ensemble des prédictions de nos modèles et de ceux de l'état de l'art.

2 État de l'art

2.1 De l'extraction... à l'abstraction : du verbe compté au verbe conté.

La disponibilité limitée de documents annotés de mots-clés a longtemps contraint les chercheurs à adopter des approches non supervisées. Cependant, la démocratisation de l'apprentissage profond a permis aux chercheurs de pleinement tirer parti des vastes corpus pour le traitement automatique des langues. Notamment, (Meng et al., 2017) introduit le travail séminal de la génération automatique de mots-clés absents à l'aide des approches neuronales de séquence-à-séquence (Sutskever et al., 2014).

1. https://anonymous.4open.science/r/egc2026kp-A908/README.md

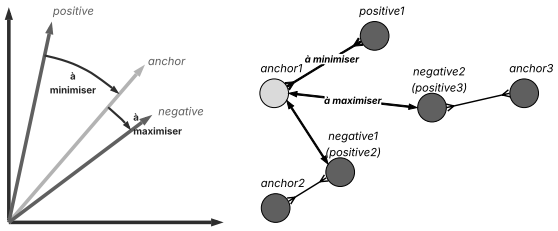

FIG. 1 – *Illustration des paradigmes TripletLoss (gauche) et MNR (droite)*

Depuis, plusieurs travaux ont exploré le réglgage-fin de modèles de langage pré-entraînés pour la génération de mots-clés. Kulkarni et al. (2022) ont montré que ces modèles, et notamment BART (Lewis et al., 2020), pouvaient être efficacement adaptés à cette tâche. Les études suivantes ont confirmé que le réglage-fin de BART constituait l'état de l'art des performances (Wu et al., 2021, 2022; Houbre et al., 2022; Boudin et Aizawa, 2024, 2025).

Cependant, la quasi popularité dont jouit BART n'est pas entièrement sans critique, (Xie et al., 2023) met en cause le manque d'adaptabilité des architectures pour transférer d'un domaine à l'autre, (Houbre et al., 2022) révèle numériquement la faible généralisation de BART aux domaines biomédical, informatique et actualités. De plus, (Boudin et Aizawa, 2025) révèle que le réglage-fin d'un BART peut prendre une dizaine à plusieurs centaines d'heures.

2.2 Modélisation : qui veut bien chercher, doit bien représenter

BERT (Devlin et al., 2019) est un réseau de neurones permettant la représentation contextuelle et bidirectionnelle de mots en vecteurs. Pour produire des représentations sémantiques comparables au niveau phrase, Sentence-BERT (sBERT) (Reimers et Gurevych, 2019) adapte BERT dans une architecture siamoise : une aggrégation génère un vecteur compact, et l'entraînement contrastif aligne les paraphrases et éloigne les antonymes.

Initialement, la fonction objectif triplet $\mathcal{L} = \max(0, ||a-p|| - ||a-n|| + m)$ consistait à minimiser la distance entre un document ancre a et un exemple similaire (positif) p tout en maximisant la distance avec un exemple non pertinent (négatif) n, avec une certaine marge m. Plus récemment, la fonction objectif *Multiple Negatives Ranking* (MNR) $\mathcal{L}_{\text{MNR}} = -\log \frac{\exp(\text{sim}(a,p)/\tau)}{\sum_{i=1}^{N} \exp(\text{sim}(a,n_i)/\tau)}$ permet d'introduire naturellement plusieurs négatifs (n_i) ce qui simplifie l'entraînement et améliore les performances (Gao et al., 2021; Choi et al., 2023).

KeyRank (Liu et Iwaihara, 2021) adapte sBERT dans un cadre triplet strictement *extractif*, où positifs et négatifs sont dérivés du document source afin d'optimiser le repérage de candidats mots-clés. À l'inverse, SimCKP (Choi et al., 2023) capitalise sur l'objectif MNR, mais demeure centré sur l'extraction et introduit un décodeur en aval de l'encodeur, le rapprochant ainsi des méthodes séquence-à-séquence. Enfin, ces deux lignes de travaux sont majoritairement évaluées sur des corpus informatique, alors que Boudin et Aizawa (2025) montrent une corrélation quasi parfaite des performances entre benchmarks du domaine, suggérant que l'effort méthodologique devrait se focaliser vers l'évaluation hors domaine et multi-domaine.

Par convention, les études ayant succédé (Meng et al., 2017) ont adopté sa définition du mot-clé absent, i.e. un mot-clé ne correspondant à aucune sous-séquence contiguë du texte

source. Ainsi, le mot-clé "droits des hommes" sera considéré absent du texte "Les hommes [...] libres et égaux en droits." du fait de l'ordre d'apparition. Du point de vue de la recherche d'information, cette définition n'est pas suffisante. En effet, la pratique en recherche d'information veut que les mots soient racinés. Par cette définition, certains mots-clés absents peuvent agir de la même manière que des mots-clés présents sur l'indexation. Du point de vue de la prédiction de mots-clés, cette définition n'est pas satisfaisante non plus. En effet, indiquer au modèle qu'un mot-clé est absent alors que les tokens qui le composent sont présents perturbe l'entraînement et pourrait expliquer leur faible performance (Gallina et al., 2020). (Boudin et Gallina, 2021) propose une extension de la définition d'un mot-clé absents en se basant sur une correspondance des tokens racinisés, on distingue alors les mots-clés :

— **P**résent : Mots-clés reflétés directement dans le document original.
— **R**éarrangé : Mots-clés composés de mots présents mais dans un ordre différent.
— **M**ixte : Mots-clés partiellement présents dans le texte.
— Inaperç**U** : Mots-clés construits à partir de mots entièrement absents du document.

Contrairement à la classification binaire prévalente, ce schéma plus précis fait une distinction entre les mots-clés qui étendent le document, i.e. "Mixte" et "Inaperçu", de ceux qui ne le font pas. Il nous permet ainsi de mieux comprendre l'impact de chaque catégorie sur l'indexation documentaire (Boudin et Gallina, 2021).

En synthèse, les approches seq2seq basées offrent d'excellentes performances intradomaine, mais souffrent (i) d'une généralisation limitée hors domaine, (ii) d'un coût d'entraînement et d'inférence élevé, et (iii) d'une difficulté à produire des mots-clés absents (classes M et U) tout en restant frugales. Ces limites invitent à explorer des paradigmes alternatifs, combinant recherche documentaire et classement sémantique.

3 Méthodologie

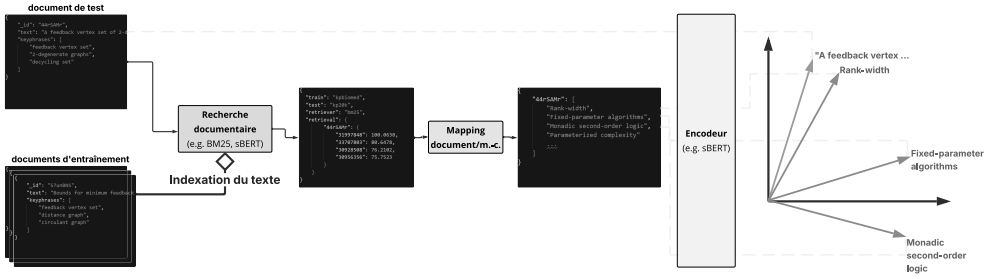

FIG. 2 – *Pipeline de SearchKeys : architecture de recherche lexicale et classement neuronal.*

Notre approche s'inscrit dans le paradigme du *retrieval-augmented ranking* (RAR). Elle repose sur l'observation que tout mot-clé pertinent pour un document apparaisse au sein d'un corpus de référence (Ye et al., 2021). Cette hypothèse fonde notre pensée en deux étapes : (i) construction d'un index où chaque document de test servira de requête ; (ii) extraction des mots-clés des documents indexés les plus proches et classement de ces candidats.

3.1 Construction d'un index inversé

À partir du corpus d'entraînement, nous construisons un index, ici avec Best Match 25 [2].

Lors de l'entraînement, nous requêtons à partir d'un document d'entraînement et récupérons les mots-clés des autres documents d'entraînement les plus proches. Cette phase agit comme construction de l'espace des candidats, transformant le problème de génération en un problème de classement.

3.2 Ranking contrastif : Sentence-Transformers fine-tuné

Les modèles sBERT (Reimers et Gurevych, 2019) ont initialement été conçus pour opérer sur des paires de longueurs comparables. Cependant, la littérature demeure lacunaire concernant les configurations asymétriques, où l'ancre (le document-requête) est substantiellement différent des candidats mots-clés en termes de longueur (Fang et al., 2024). Cette asymétrie induit des biais de normalisation et de calibration des similarités qui dégradent les performance de classement (Fayyaz et al., 2025). Ceci nous invite à affiner un sBERT.

L'objectif *Multiple Negatives Ranking* (MNR) exploite efficacement les négatifs présents dans un batch : chaque paire (d_i, p_i^+) considère tous les autres candidats $\{p_k^+\}_{k\neq i}$ du batch comme négatifs pour d_i, sans nécessité de sélection explicite à l'instar du triplet loss. Toutefois, cette hypothèse est inadéquate dans notre contexte. Un même mot-clé peut en effet être pertinent pour plusieurs documents.

Pour corriger ce biais, nous introduisons le *Masked Multiple Negatives Ranking* (mMNR) : neutraliser, pour chaque document, les candidats qui sont également des positifs au-dit document. Soit un batch de documents $\{d_1, \ldots, d_N\}$ et un ensemble d'annotations $\mathcal{A}$ définissant les couples positifs (d_i, p_i^+). Nous calculons la matrice des similarités $\mathbf{S} \in \mathbf{R}^{N\times M}$ entre les N documents et les M candidats, puis appliquons un masque défini par $\tilde{\mathbf{S}}_{ij} = \mathbf{S}_{ij}$ si $(d_i, p_j) \notin \mathcal{A}$ et $-\infty$ sinon. Ainsi, pour le document d_i et son mot-clé de référence p_i^+, la perte mMNR s'écrit $\ell_i = -\log\left(\frac{\exp(\mathbf{S}_{i,+}/\tau)}{\sum_{j:\,(d_i,p_j)\notin\mathcal{A}}\exp(\tilde{\mathbf{S}}_{ij}/\tau)}\right)$, où τ est la température et $+$ désigne le candidat positif de d_i. En supprimant la contribution des faux négatifs, mMNR réaligne correctement l'objectif contrastif et améliore la stabilité de l'entraînement. Considérons un batch minimal où d_1 est annoté par $\{p_1, p_2\}$ et d_2 par $\{p_2\}$:

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \xrightarrow[\text{mask}]{} \begin{bmatrix} s_{11} & -\infty \\ s_{21} & s_{22} \end{bmatrix} \xrightarrow[\text{softmax}]{} \mathcal{L} = \begin{bmatrix} -\log\frac{e^{s11}}{e^{s11}+e^{-\infty}} & 0 \\ \ldots & \ldots \end{bmatrix}$$

4 Evaluation

4.1 Jeux de données

Afin d'évaluer notre approche dans un contexte multi-domaine, nous reprenons le benchmark dans (Houbre et al., 2022) :

— **kpbiomed** (Houbre et al., 2022) : notices bibliographiques (titres et résumés) d'articles médicaux de PubMed annotés par leurs auteurs sans vocabulaire contrôlé.

2. https://github.com/xhluca/bm25s (paramètres par défaut `k1=1.5, b=0.75`)

— **kptimes** (Gallina et al., 2019) : articles de journaux (New York Times, Japan Times), annotés par des éditeurs professionels
— **kp20k** (Meng et al., 2017) : métadonnées de conférences en informatique issus de "*various online digital libraries, including ACM Digital Library, ScienceDirect etc.*"

	Document		**Mots-clés**				**Distribution (%)**			
	$\lvert\mathcal{D}\rvert$	**Tokens**	$\lvert\mathcal{V}\rvert$	$\mathcal{C}$	**Tokens**	d	**P**	**R**	**M**	**U**
kpbiomed										
train	500k	342_{107}	822.7k	5_{2}	6_{2}	3_{22}	67.53	7.33	11.69	13.45
dev	20k	343_{108}	66.9k	5_{3}	6_{2}	2_{3}	67.48	7.19	11.54	13.79
test	20k	343_{107}	66.6k	5_{2}	6_{2}	2_{3}	67.50	7.23	11.73	13.54
total	540k	342_{107}	870.8k	5_{2}	6_{2}	3_{23}	67.53	7.32	11.68	13.46
kp20k										
train	530.8k	205_{85}	723.6k	5_{4}	5_{1}	4_{42}	63.53	9.87	15.09	11.51
dev	20k	206_{85}	56.8k	5_{4}	5_{1}	2_{5}	63.47	9.91	15.04	11.58
test	20k	206_{86}	57.4k	5_{4}	5_{1}	2_{5}	63.63	9.91	14.93	11.53
total	570.8k	205_{85}	760.8k	5_{4}	5_{1}	4_{44}	63.53	9.87	15.08	11.51
kptimes										
train	259.9k	463_{115}	105.9k	5_{2}	5_{1}	12_{115}	50.06	17.36	24.63	7.95
dev	10k	463_{114}	13.8k	5_{2}	5_{1}	4_{12}	49.57	17.65	24.64	8.15
test	20k	456_{107}	23.0k	5_{2}	5_{1}	4_{20}	66.22	10.00	14.06	9.72
total	289.9k	462_{115}	119.6k	5_{2}	5_{1}	12_{117}	51.16	16.87	23.90	8.08

TAB. 1 – *Statistiques sur les jeux de données. Les tokens des documents sont mesurés avec* `bart-base`*. La notation X_Y indique une valeur moyenne X et un écart-type Y, i.e. $X \pm Y$.*

Les statistiques de ces trois corpus sont synthétisées dans la Table 1. On remarque que **kptimes** présente un volume de documents $|\mathcal{D}|$ deux fois inférieur à **kpbiomed** et **kp20k** mais deux fois plus long, dépassent la limite de longueur de 512 tokens couramment employée. Aussin les vocabulaires $|\mathcal{V}|$ de **kpbiomed** et **kp20k** sont 6 à 7 fois plus grands que **kptimes**, et leurs mots-clés sont en moyenne d 4 fois moins assignés.

4.2 Expérimentations

Nous réglons un encodeur avec `sentence-transformers` en partant du modèle all-MiniLM-L6-v2 et en optimisant l'objectif présenté en Éq. 3.2. L'entraînement est réalisé avec une taille de lot de 1024 sur une machine équipée d'un accélérateur **NVIDIA L40S Tensor Core** (1 × L40S ; 362 TFLOPs en `fp16`, 48 Go de mémoire GPU). Le nœud est loué auprès de `lightning.ai` au coût indicatif de 2,89 $/h. L'apprentissage a duré **17 heures**.

En inférence, nous indexons les trois domaines, interrogeons l'index avec l'ensemble de test, puis effectuons un ranking des candidats collectés auprès des **5** documents les plus proches. L'ensemble de ce pipeline d'inférence illustré en Figure 2 s'exécute en **1 minute 45 secondes**.

Pour situer notre approche, nous retenons plusieurs systèmes *seq2seq* fondés sur BART et réglés à chaque domaine considéré : **BART-base-KP20k** (Boudin et Aizawa, 2025) correspondant à BART-base *fine-tuné* sur `kp20k` (15 époques, 63 heures sur 1 × NVIDIA RTX 2080 Ti). **BioBART-small** (Houbre et al., 2022) correspondant à BioBART-base (Yuan et al., 2022) (pré-

réglé au domaine biomédical), puis *fine-tuné* à la génération de mots-clés sur `kpbiomed`. (10 époques, 9 heures sur 4 × Tesla V100 32 Go). **BART-kptimes** (Houbre et al., 2022) avec peu de détails publics ; nous supposons des conditions expérimentales analogue à **BioBART-small**.

Face à la faible généralisation de ces modèles hors-domaine (Houbre et al., 2022), nous introduisons **BART-KP**, obtenu par *fine-tuning* de `BART-base` sur les trois jeux de données étudiés. L'entrée correspond à la concaténation `title<s>abstract` (Ye et al., 2021; Chowdhury et al., 2022; Chen et al., 2019) ; la sortie suit le paradigme *One2Seq* (Yuan et al., 2020), i.e. concaténation des mots-clés séparés par le jeton spécial `<SEP>` et ordonnés par catégories *présent→absent*, qui constitue le cadre d'apprentissage optimal (Meng et al., 2021).

L'entraînement, réalisé sur la même infrastructure que SEARCHKEYS, s'est déroulé sur 10 époques pendant 16 heures avec un taux d'apprentissage $\lambda = 5 \times 10^{-5}$ (optimiseur ADAMW) et une taille de lot de 168. L'inférence requiert 1 heure avec une recherche en faisceau `num_beams=5`. Ce paramétrage se justifie à double titre : (i) la recherche en faisceau améliore la génération de mots-clés absents (Meng et al., 2021) ; (ii) on aligne l'inférence BART à notre cadre de récupération des cinq voisins les plus proches de SEARCHKEYS, en l'assimilant à la génération des cinq séquences de mots-clés les plus probables.

D'autres architectures proches, telles que One2Set (Ye et al., 2021) ou KG-KE-KR-M (Chen et al., 2019), n'ont pas été entraînées au-delà du domaine scientifique (`kp20k`) et nécessiteraient un fine-tuning séparé pour chaque domaine, puis combiné sur les trois corpus réunis. Un tel plan expérimental, bien que pertinent, dépasse le cadre de cette étude. Nous concentrons donc nos efforts comparatifs sur un unique décodeur `BART-KP` multi-domaine, qui sert de référence seq2seq forte face à SEARCHKEYS.

4.3 Métriques

L'évaluation de la prédiction de mots-clés compare les références $\mathcal{Y} = \{y_1, \ldots, y_N\}$ aux prédictions $\widehat{\mathcal{Y}} = \{\hat{y}_1, \ldots, \hat{y}_K\}$ via la `f-score` et le `rappel`. La `f-score` est couramment utilisée pour l'*extraction*, tandis que le `rappel` est privilégié pour l'*abstraction*. Afin d'assurer une comparaison équitable, les prédictions sont *tronquées* au nombre de références (par classe PRMU) du document, noté $@\mathcal{O}$ (Ye et al., 2021). Avant comparaison nous racinisions (*stemming*) avec `PorterStemmer`. Les résultats sont établis dans la Table 2 et les caractéristiques de l'expérience dans la Table 3.

5 Discussions

5.1 Généralisation inter-domaine : le défi fondamental des architectures seq2seq

La littérature récente, notamment Houbre et al. (2022) et Xie et al. (2023), a établi que les modèles *seq2seq* spécialisés souffrent d'une dégradation drastique lorsqu'ils opèrent hors de leur domaine d'entraînement. Nos expériences confirment et approfondissent ce constat.

Notre baseline multi-domaine **BART-KP**, entraîné sur l'ensemble des trois corpus pour une durée équivalente aux modèles spécialisés, révèle une incapacité structurelle à naviguer la diversité inter-domaine. Sur les mots-clés **présents**, il demeure systématiquement inférieur

		bart-kpbiomed	bart-kp20k	bart-kptimes	bart-KP	SearchKeys
kpbiomed	*Présent*	**0.394**	0.334	0.130	0.226	0.324† (x1.43)
	Réarrangé	0.012	0.005	0.001	0.001	**0.031**† (x31)
	Mixte	0.010	0.002	0.001	0.001	**0.036**† (x36)
	Inaperçu	0.017	0.004	0.003	0.001	**0.057**† (x57)
kp20k	*Présent*	0.303	0.348	0.046	0.237	**0.371**† (x1.57)
	Réarrangé	0.011	0.021	0.001	0.004	**0.085**† (x21.25)
	Mixte	0.004	0.013	0.001	0.002	**0.095**† (x47.5)
	Inaperçu	0.003	0.005	0.002	0.001	**0.080**† (x80)
kptimes	*Présent*	0.253	0.215	**0.473**	0.370	0.459† (x1.24)
	Réarrangé	0.003	0.004	0.147	0.067	**0.167**† (x2.49)
	Mixte	0.003	0.002	0.172	0.082	**0.205**† (x2.50)
	Inaperçu	0.015	0.009	0.096	0.028	**0.120**† (x4.29)
total	*Présent*	0.317	0.299	0.216	0.278	**0.385**† (x1.38)
	Réarrangé	0.009	0.010	0.050	0.024	**0.094**† (x3.92)
	Mixte	0.006	0.006	0.058	0.028	**0.112**† (x4)
	Inaperçu	0.012	0.006	0.034	0.010	**0.086**† (x8.6)

TAB. 2 – *Performances sur les classes PRMU (F-Score@$\mathcal{O}$ présents; Rappel@$\mathcal{O}$ absents).* ***Meilleur***, *second*. † *: gain signif. ($\alpha = .05$) de SearchKeys sur BART; $\times$: ratio.*

aux spécialistes : *kpbiomed* (**0.226** vs **BioBART-small 0.394**), *kp20k* (**0.237** vs **BART-base-KP20k 0.348**), *kptimes* (**0.370** vs **BART-kptimes 0.473**). Cette dégradation découle d'un « bruit d'apprentissage » : les distributions lexicales et sémantiques incompatibles entre domaines créent des objectifs conflictuels, forçant le modèle à un compromis l'empêchant d'exceller nulle part. Les **absents** subissent une pénalité encore plus sévère, avec des écarts massifs (p.ex., classe *U* sur *kpbiomed* : **0.001** vs **0.017** pour BioBART-small).

C'est précisément sur ce défi que **SearchKeys** déploie une stratégie radicalement différente. Pour un temps d'entraînement quasi identique (**17.3 h** vs **16.5 h**), il *dépasse la majorité des baselines sur l'ensemble des axes critiques*. Globalement, SearchKeys atteint **0.385** en mots-clés présents (vs **0.278** pour BART-KP) et domine sur **tous** les absents : *Inaperçu* (*U* **0.086**), *Mixte* (*M* **0.112**), *Réarrangé* (*R* **0.094**), avec des gains massifs attestant une amélioration **8.6 fois** sur *U* comparé à BART-KP. L'approche par recherche et classement transforme le défi : au lieu de forcer un seul décodeur à maîtriser plusieurs domaines, elle puise dans un index unifié pour construire un espace de candidats pertinents que le classeur apprend à ordonner, évitant les hallucinations courantes aux architectures génératives.

SearchKeys reste compétitif même face aux spécialistes mono-domaine, gardant un net avantage sur les absents. Sur les présents, il est second ou *au contact* seulement dans les cas où l'expressivité séquentielle des décodeurs conserve historiquement l'avantage (p.ex., *kptimes–Présent* 0.459 vs 0.473). Cette légère différence s'explique par la nature de kptimes : un vocabulaire restreint et une forte réutilisation des mots-clés permettent aux modèles hyper-spécialisés de mémoriser et reproduire aisément des termes fréquents. De même, sur *kpbiomed-Présent*, SearchKeys (**0.324**) demeure inférieur à BioBART-small (**0.394**), mais reste au contact de BART-base-KP20k (**0.348**), ce dernier étant lui aussi un BART réentraîné

Approche / Étape	Temps (h)	Énergie (kWh)	CO_2e (kg)	Coût ($)
SearchKeys (all-MiniLM-L6-v2)				
(Pré-)Entraînement				
Tokenisation — 1,34M docs.	0.030	0.011	0.004	0.087
Indexation BM25 — 1,34M docs.	0.013	0.005	0.002	0.038
Recherche BM25 — 1,34M docs.	0.408	0.143	0.057	1.179
Réglage fin — 6,7M paires	17.270	6.045	2.419	49.911
Inférence				
Recherche BM25 — 60k docs.	0.018	0.006	0.003	0.052
Encodage — 60k docs.	0.009	0.003	0.001	0.026
Encodage — 142k candidats	0.001	0.000	0.000	0.003
Classement des candidats	0.001	0.000	0.000	0.003
Total	**17.750**	**6.213**	**2.486**	**51.300**
BART-KP (BART-base)				
Réglage fin — 1,29M paires	16.533	5.786	2.314	47.780
Inférence (60k documents)	0.800	0.280	0.112	2.312
Total	**17.333**	**6.066**	**2.426**	**50.092**

TAB. 3 – *Temps, énergie, empreinte carbone et coût de calcul par étape sur L40S.*

sur son domaine de spécialité. Cette hiérarchie révèle que les modèles déjà fine-tunés sur une architecture de base partagée bénéficient d'une avance initiale que le classement sémantique peine à compenser dans les cas hautement spécialisés. Néanmoins, même face aux spécialistes, SearchKeys domine sur les absents (respectivement R/M/U), mettant en évidence où les bénéfices de notre approche RAR résident vraiment : l'enrichissement sémantique par découverte de candidats absents du document source mais présents chez ses voisins les plus proches.

5.2 Au-delà du temps d'entraînement : l'hégémonie neuronale justifie-t-elle son coût opérationnel ?

L'hégémonie des modèles neuronaux génératifs repose rarement sur une remise en question de son coût réel au-delà du simple temps d'entraînement. Bien que les deux approches partagent un investissement d'entraînement équivalent, l'écart en inférence redéfinit le calcul d'efficacité.

Pour 60 000 documents, BART-KP avec recherche en faisceau de 5 requiert **48 minutes** (soit 0.048 s/document), tandis que l'intégralité du pipeline SEARCHKEYS — requête BM25, encodage des documents et candidats, classement par similarité — s'exécute en seulement **105 secondes** (**~1 min 45**), soit **27 fois plus rapide**. Cet écart provient de deux facteurs : (i) SEARCHKEYS n'effectue qu'un seul encodage sBERT par document, tandis que BART génère itérativement 5 séquences ; (ii) la projection vectorielle et le tri par similarité surpassent le décodage autorégressif, en particulier grâce à l'architecture parallèle. **Côté mémoire**, l'index BM25 pèse **~2.1 GB** et reste *inerte* : il se réplique et se shard facilement sur un stockage bon marché (objet/SSD), se met en cache et ne nécessite aucun GPU. À l'inverse, **BART-KP** n'occupe que **406 MB** sur disque, mais doit rester *chargé et actif en GPU* à l'inférence : le décodage

par faisceau maintient les poids *et* des états intermédiaires (p. ex. KV-cache/activations), mobilisant plusieurs Go de VRAM et contraignant la taille de lot.

Les implications cascadent sur l'infrastructure. La consommation énergétique chute de **0.280 kWh** pour BART à **0.009 kWh** pour SearchKeys ; l'empreinte CO_2e passe de **0.112 kg** à **0.004 kg** ; le coût se divise par 44 (**$2.312** vs **$0.084**). Extrapolé à une indexation annuelle massifiée, par exemple 35 millions d'articles biomédicaux (PubMed), BART impliquerait des milliers d'heures GPU, contre minutes pour SearchKeys. Cette sobriété computationnelle rend l'approche *scalable* et durable sans sacrifier la performance, questionnant la justification de la génération coûteuse pour des gains souvent marginaux dans les cas intra-domaine.

5.3 Limitations et perspectives futures

SearchKeys ne peut classer que les mots-clés préalablement récupérés. Cette dépendance pose problème dans deux cas : (i) les termes ultra-rares ou émergents absents du corpus d'entraînement ; (ii) les domaines très spécialisés ou en évolution rapide générant un vocabulaire entièrement neuf. Bien que nos résultats montrent une robustesse acceptable en classe Inaperçu (**0.086**), les améliorations authentiques restent hors d'atteinte. *Perspectives* : hybrider RAR avec génération légère pour les cas rares, intégrer dense retrievers et cross-collection indexing, ou procéder à un rafraîchissement incrémental de l'index pour suivre l'actualité terminologique.

Les métriques d'évaluation en génération de mots-clés héritent d'une *référence humaine* intrinsèquement subjective. Le problème s'amplifie pour les mots-clés **absents** (notamment `M/U`) dont le lexique varie fortement selon l'annotateur. Cela a pour deux conséquences ; (i) un modèle peut produire un mot-clé *pertinent* mais "faux" au sens strict de la référence, (ii) l'optimisation risque d'aligner le système sur des biais d'annotateur. Explorer le potentiel des mots-clés pour des tâches utiles (Xie et al., 2023), comme la **recherche documentaire** (Boudin et Gallina, 2021) représente un enjeu conséquent et cohérent pour ce domaine de recherche.

Références

Boudin, F. et A. Aizawa (2024). Unsupervised domain adaptation for keyphrase generation using citation contexts. In *Findings of the Association for Computational Linguistics : EMNLP 2024*.

Boudin, F. et A. Aizawa (2025). An analysis of datasets, metrics and models in keyphrase generation. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*.

Boudin, F. et Y. Gallina (2021). Redefining absent keyphrases and effect on retrieval. In *NAACL 2021*.

Chen, W., H. P. Chan, P. Li, L. Bing, et I. King (2019). An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. In *EMNLP-IJCNLP 2019*.

Choi, M., C. Gwak, S. Kim, S. Kim, et J. Choo (2023). SimCKP : Simple contrastive learning of keyphrase representations. In *Findings of the Association for Computational Linguistics : EMNLP 2023*.

Chowdhury, M. F. M., G. Rossiello, M. Glass, N. Mihindukulasooriya, et A. Gliozzo (2022). Applying a generic sequence-to-sequence model for simple and effective keyphrase generation.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *EMNLP-IJCNLP 2019*.

Fang, Y., J. Zhan, Q. Ai, J. Mao, W. Su, J. Chen, et Y. Liu (2024). Scaling laws for dense retrieval. In *47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Fayyaz, M., A. Modarressi, H. Schuetze, et N. Peng (2025). Collapse of dense retrievers : Short, early, and literal biases outranking factual evidence. In *63rd Association for Computational Linguistics*.

Gallina, Y., F. Boudin, et B. Daille (2019). KPTimes : A large-scale dataset for keyphrase generation on news documents. In *12th International Conference on Natural Language Generation*.

Gallina, Y., F. Boudin, et B. Daille (2020). Large-scale evaluation of keyphrase extraction models. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*.

Gao, T., X. Yao, et D. Chen (2021). SimCSE : Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Houbre, M., F. Boudin, et B. Daille (2022). A large-scale dataset for biomedical keyphrase generation. In *13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*.

Kulkarni, M., D. Mahata, R. Arora, et R. Bhowmik (2022). Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics : NAACL 2022*.

Kwon, S. (2018). Characteristics of interdisciplinary research in author keywords appearing in journals.

Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, et L. Zettlemoyer (2020). BART : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL 2020*.

Liu, T. et M. Iwaihara (2021). Embedding-based supervised learning for keyphrase extraction. In *Proceedings of DEIM Forum 2021 (H21-3)*.

Meng, R., S. Zhao, S. Han, et D. He (2017). Deep keyphrase generation. In *ACL 2017*.

Meng, R., S. Zhao, A. Trischler, et D. He (2021). Empirical study keyphrase generation. In *NAACL 2021*.

Reimers, N. et I. Gurevych (2019). Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Strader, C. (2009). Author-assigned keywords versus library of congress subject headings implications for the cataloging of electronic theses and dissertations.

Sutskever, I., O. Vinyals, et Q. V. Le (2014). Sequence to sequence learning with neural networks.

Wu, D., W. Ahmad, S. Dev, et K.-W. Chang (2022). Representation learning for resource-constrained keyphrase generation. In *Findings of the Association for Computational Linguistics : EMNLP 2022*.

Wu, H., W. Liu, L. Li, D. Nie, T. Chen, F. Zhang, et D. Wang (2021). UniKeyphrase : A unified extraction and generation framework for keyphrase prediction. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*.

Xie, B., J. Song, L. Shao, S. Wu, X. Wei, B. Yang, H. Lin, J. Xie, et J. Su (2023). From statistical methods to deep learning, automatic keyphrase prediction : A survey.

Ye, J., T. Gui, Y. Luo, Y. Xu, et Q. Zhang (2021). One2Set : Generating keyphrases as a set. In *ACL 2021*.

Yuan, H., Z. Yuan, R. Gan, J. Zhang, Y. Xie, et S. Yu (2022). BioBART : Pretraining and evaluation of a biomedical generative language model. In *21st Workshop on Biomedical Language Processing*.

Yuan, X., T. Wang, R. Meng, K. Thaker, P. Brusilovsky, D. He, et A. Trischler (2020). One size does not fit all : Generating and evaluating variable number of keyphrases. In *ACL 2020*.

Summary

Keyphrase generation remains a challenge for current state-of-the-art methods, which remain heavily centered on neural encoder-decoder architectures. These models often struggle to generalize across domains, or even to generate satisfactory absent keyphrases within their own training domain. Moreover, they require substantial computational resources for marginal performance gains.

Our approach introduces an encoder-only architecture dedicated to ranking keyphrases drawn from cross-domain candidate pools. Using the same training set as the decoders but treating it as an index, each test document is processed as a query: we retrieve keyphrases from its nearest neighbors and learn to rank them with a fine-tuned Sentence-Transformer adapted to multiple domains.

Training relies on a multiple negatives ranking objective, where each test document is paired with its reference keyphrase among a shared pool of negative candidates. However, since certain keyphrases may be relevant to multiple documents, we mask their loss contributions to avoid penalizing the model for potentially valid associations. This adaptation better captures semantic overlaps between documents.

We compare our method against strong seq2seq baselines, evaluating f-score and recall for present and absent keyphrases, out-of-domain robustness, latency, training and inference costs, as well as environmental footprint. Our retrieval-ranking approach matches or surpasses generative baselines while significantly mitigating the limitations inherent to neural decoder architectures. This work thus positions itself as an alternative to seq2seq models for document indexing in knowledge bases using keyphrases.