

# Extraction de mots-clés à partir d'articles scientifiques: comparaison entre modèles traditionnels et modèles de langue

Nacef BENMANSOUR, Motasem ALRAHABI

Sorbonne Université

[prenom-nom@sorbonne-universite.fr](mailto:prenom-nom@sorbonne-universite.fr)

## Résumé

L'extraction automatique des mots-clés est essentielle pour condenser le contenu des documents, améliorer la recherche d'informations et analyser les tendances. Traditionnellement, cette tâche utilise des méthodes statistiques comme TF-IDF et des modèles d'apprentissage automatique. Cependant, les modèles de langage de grande taille (LLMs) comme GPT-4 offrent une compréhension plus contextuelle et nuancée des textes. Cette étude compare les approches traditionnelles et modernes d'extraction de mots-clés en utilisant des articles scientifiques de la base de données HAL. Les résultats montrent que les LLMs surpassent les méthodes traditionnelles en termes de précision et de pertinence, suggérant des améliorations pour les systèmes de recherche scientifique.

## Mots-clés

Extraction de mots-clés, modèles de langage, évaluation, corpus HAL.

## 1) Contexte

Le présent travail s'inscrit dans un projet plus vaste visant à recenser les compétences et l'expertise des laboratoires et des chercheurs de Sorbonne Université, afin de mieux les valoriser, tout en facilitant les partenariats et la formation dans le cadre de projets scientifiques. Dans le contexte de la montée en puissance de la science ouverte, ce projet interdisciplinaire repose sur la collecte et la structuration des informations, ainsi que sur le développement d'outils et d'interfaces dédiées à la recommandation et à l'interrogation de données. L'évaluation de l'extraction automatique des mots-clés, essentielle pour structurer et rendre accessible l'information, constitue le cœur de cette étude, initiée dans le cadre d'un stage universitaire à Sorbonne Université.

## 2) Introduction

L'extraction de mots-clés est une étape cruciale pour identifier les termes les plus pertinents et représentatifs d'un document ou d'un corpus, facilitant la compréhension et l'exploitation de l'information contenue. Cette technique possède un large éventail d'applications, allant de la condensation de contenu et de l'optimisation des moteurs de recherche à l'extraction d'informations, au résumé automatique et à la détection des tendances émergentes. Les méthodes traditionnelles d'extraction, basées sur des règles ou des calculs quantitatifs, montrent cependant des limites dans la capture des subtilités sémantiques et contextuelles des mots-clés. L'avènement des modèles d'apprentissage profond a permis le développement de nouvelles approches qui allient la rigueur des méthodes statistiques à la capacité des LLMs à interpréter le contexte et la sémantique des mots ([Song et al., 2023](#)).

Ce travail propose, après un état des lieux des recherches récentes, de comparer les performances des approches traditionnelles et des techniques modernes dans le domaine de l'extraction de mots-clés.

### 3) Approches pour l'extraction de mots-clés

L'évolution des approches pour l'extraction de mots-clés et de phrases-clés peut être divisée en plusieurs étapes, reflétant les progrès technologiques et méthodologiques dans le traitement du langage naturel:

- Règles linguistiques: Ces méthodes se basent sur l'analyse syntaxique, comme l'extraction de syntagmes nominaux ou l'analyse des n-grammes. Des algorithmes comme YAKE! ([Campos et al., 2020](#)) utilisent des caractéristiques linguistiques comme la place d'un mot dans la phrase pour extraire les mots-clés.
- Approches statistiques: Certaines méthodes comme TF-IDF attribuent un poids à chaque mot en fonction de sa fréquence d'apparition dans le texte et de sa distribution dans un corpus plus large ([Salton & Buckley, 1988](#)). L'algorithme RAKE, par exemple, identifie les mots-clés en fonction de leur co-occurrence dans les phrases et leur importance relative dans le texte. Les n-grammes permettent également d'identifier des séquences de mots fréquentes ([Justeson & Katz, 1995](#)). D'autres approches exploitent la position des mots dans les phrases pour identifier les termes clés ([Turney, 2000](#)) ou se basent sur des distributions statistiques plus complexes ([Church & Gale, 1995](#)).
- Apprentissage supervisé: Ces méthodes supervisées, comme les algorithmes de classification, exploitent des données annotées pour entraîner des modèles à extraire des mots-clés. KP-Miner ([El-Beltagy, 2009](#)) et les approches supervisées de ([Papagiannopoulou et al., 2020](#)) en sont des exemples.
- Apprentissage non supervisé: Certaines méthodes non supervisées s'appuient sur l'utilisation de graphes. Des méthodes comme TextRank ([Mihalcea & Tarau, 2004](#)), SingleRank ([Wan & Xiao, 2008](#)), et MultipartiteRank ([Boudin, 2018](#)) utilisent des graphes de co-occurrence de mots pour extraire les mots-clés sans avoir besoin de données annotées. TopicRank ([Bougouin et al., 2013](#)) et PositionRank ([Florescu & Caragea, 2017](#)) exploitent également des graphes pour évaluer l'importance des mots. Ces méthodes, bien que robustes et largement utilisées, peuvent manquer de contextualisation et de nuance dans l'extraction des mots-clés.
- Vectorisation: Certaines méthodes ont exploité des représentations vectorielles plus riches des mots ou des phrases pour l'extraction de mots-clés. Par exemple, EmbedRank utilise deux variantes : l'une basée sur Word2Vec ([Mikolov et al., 2013](#)) pour créer des embeddings des phrases candidates et du document, et l'autre s'appuyant sur Sent2Vec ([Pagliardini et al., 2017](#)) pour générer des embeddings plus riches des phrases. Les phrases candidates sont ensuite extraites à l'aide de motifs syntaxiques, tels que les Part of Speech (PoS), et classées en fonction de leur similarité cosinus avec le document ([Bennani-Smires et al., 2018](#)). Dans la continuité de ces travaux, des approches plus récentes, telles que PatternRank et KeyBERT, ont intégré des embeddings contextuels et des modèles de langage avancés, tels que SBERT ou BERT, pour améliorer l'évaluation des mots-clés. Ces méthodes combinent également des motifs syntaxiques, comme les PoS, afin d'identifier les termes clés et évaluent leur proximité contextuelle avec le texte source ([Schopf et al., 2023](#) ; [Grootendorst, 2020](#)). Ces méthodes ont marqué une transition importante vers des approches plus contextuelles, tout en restant en deçà des capacités des LLMs.

- Les approches génératives reposent sur les réseaux neuronaux profonds et les LLMs tels que les familles BERT, GPT et LLaMA. Ces modèles, grâce à leur capacité à saisir des relations contextuelles complexes entre les termes, permettent une extraction précise et pertinente des mots-clés via des approches zero-shot, few-shot ou par fine-tuning. Ils utilisent des réseaux de neurones basés sur une architecture novatrice, les Transformers ([Vaswani et al., 2017](#)), pour comprendre le contexte et générer des mots-clés pertinents basés sur une compréhension plus profonde du texte. Ces modèles peuvent être utilisés directement pour extraire des mots-clés, ou bien être fine-tunés sur des tâches spécifiques d'extraction de mots-clés avec des données annotées dans un domaine particulier. Contrairement aux méthodes extractives, qui se limitent à sélectionner des mots-clés directement présents dans le texte, les approches génératives modernes peuvent créer ou reformuler des mots-clés, même s'ils n'apparaissent pas explicitement dans le texte d'origine.

D'autres tendances émergentes incluent l'intégration d'approches hybrides, l'évolution de l'apprentissage automatique non supervisé, l'adaptation de modèles pré-entraînés pour des domaines spécifiques ou des langues moins courantes ([Song et al., 2023](#)), ainsi que l'extraction thématique de mots-clés à l'aide de techniques comme BERTopic pour regrouper les documents selon leurs thèmes centraux ([Grootendorst, 2020](#)).

Soulignons enfin que des bibliothèques Python comme NLTK (<https://www.nltk.org/>), Scikit-learn (<https://scikit-learn.org/>), et Spacy (<https://spacy.io/>) fournissent des modules pour l'extraction de mots-clés, avec des outils tels que TF-IDF et des méthodes basées sur l'analyse linguistique. Gensim (<https://github.com/piskvorky/gensim>) est également couramment utilisée pour la modélisation de sujets et l'extraction de termes clés. Par ailleurs, la bibliothèque PKE (<https://github.com/boudinfl/pke>) implémente plusieurs algorithmes avancés, offrant une large couverture de techniques basées sur des graphes et des statistiques.

## 4) Méthodologie

La méthodologie que nous avons employée pour comparer différentes méthodes d'extraction des mots-clés comprend plusieurs étapes :

- Préparation des données: Extraction des résumés et des mots-clés d'auteurs à partir d'articles scientifiques de la base de données HAL (<https://hal.science/>). Les expériences ont été menées en deux configurations : une avec les résumés et les titres combinés, et une autre avec uniquement les résumés, sans les titres.
- Mise en œuvre des approches : Nous avons évalué différentes méthodes d'extraction de mots-clés, en les répartissant en deux catégories : les méthodes génératives modernes et les autres approches, qualifiées de traditionnelles. Pour l'utilisation des LLMs, cette étude a adopté une approche zero-shot, choisie pour sa simplicité d'implémentation et son efficacité, tout en reflétant les performances de base représentatives de l'état de l'art pour ces modèles.
- Évaluation: Les mots-clés extraits sont évalués en fonction de leur pertinence et de leur précision par rapport aux mots-clés fournis par les auteurs dans leurs articles. Les critères d'évaluation incluent la précision, le rappel et la mesure F1.
- Analyse des résultats: Comparaison des performances des méthodes traditionnelles et modernes à l'aide de statistiques descriptives et d'analyses qualitatives.

Ces étapes permettent d'assurer une comparaison robuste pour observer les divergences de performance entre approches traditionnelles et modernes.

## 5) Les données

Les données utilisées pour cette étude proviennent de la plateforme HAL, une archive ouverte dédiée à la diffusion des travaux de recherche scientifique. Des travaux récents, tels que HALvest ([Kulumba et al., 2024](#)), démontrent le potentiel sous-exploité de la base HAL pour l'exploration et l'analyse des publications scientifiques.

Les articles sélectionnés pour les chercheurs de SU couvrent divers domaines scientifiques. Chaque article est accompagné de résumés et de mots-clés fournis par les auteurs, qui serviront de référence pour évaluer la qualité des méthodes d'extraction. Pour une première exploration, notre jeu de données contient les articles d'une centaine de chercheurs affiliés au Sorbonne Center for Artificial Intelligence (SCAI). Ce jeu de données a été constitué à l'aide d'un script exploitant l'API de HAL. Les données collectées comptaient environ 12 000 articles. Un premier tri a permis d'éliminer 1 300 doublons, tandis qu'environ 6 000 autres articles ont été exclus en raison de l'absence de mots-clés ou de résumés. Après ce filtrage, le corpus final se compose de 4 700 articles exploitables pour notre analyse, représentant environ 30 % des données initiales.

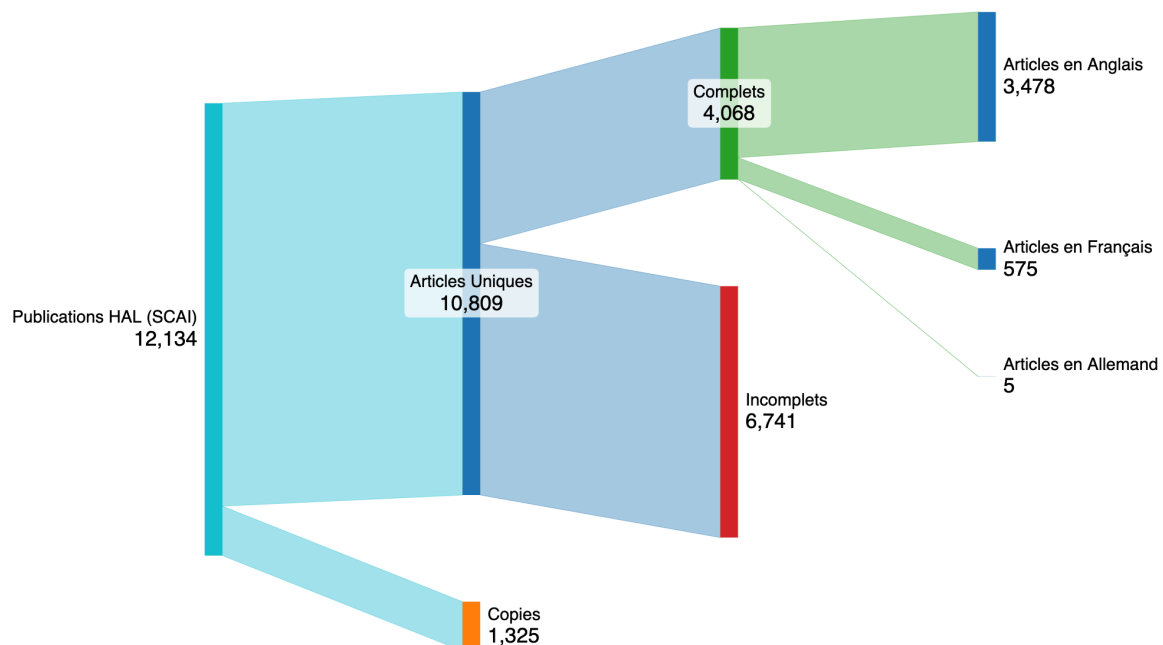


Figure 1: Répartition des articles selon la langue

Une première observation révèle une répartition linguistique marquée avec 85 % des articles en anglais, contre 15 % en français. Concernant les articles en anglais, la moyenne du nombre de mots-clés par article est de 5,35 avec une longueur moyenne des mots-clés de 2,14 mots. En comparaison, pour les articles en français, le nombre moyen de mots-clés est légèrement supérieur à 6,32, avec une longueur moyenne de 2,23 mots.

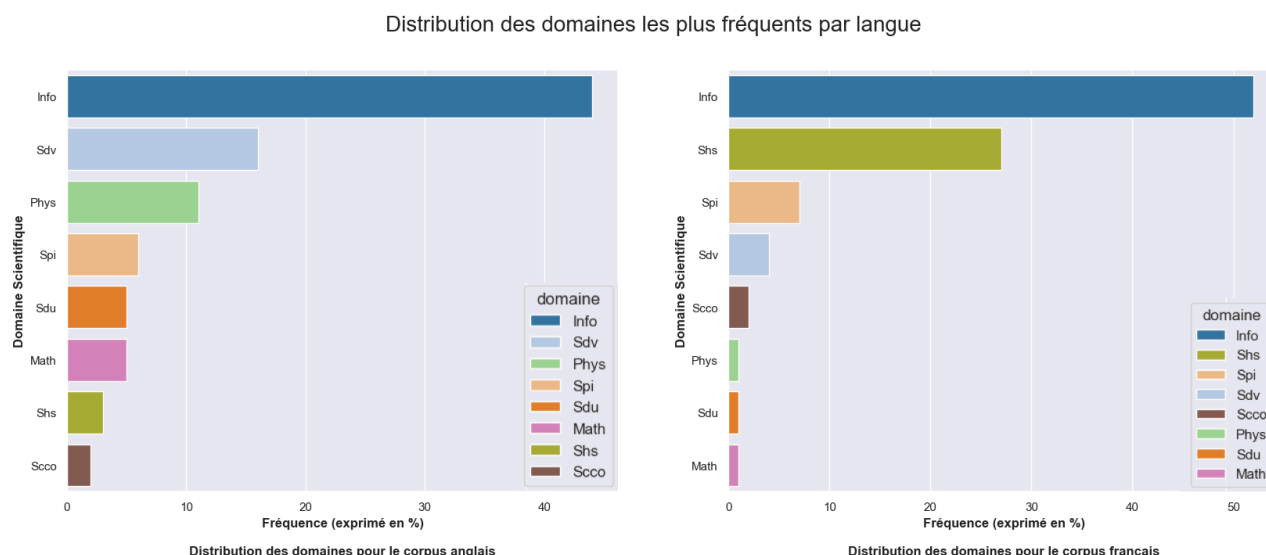


Figure 2: Distribution des domaines selon la langue

La distribution des domaines scientifiques varie également selon la langue, comme illustré dans la figure 2. Sans surprise, le domaine de l'informatique reste majoritaire pour les deux langues. Les sciences humaines arrivent en deuxième position en français, tandis que les sciences de la vie prennent cette place en anglais. Les sciences humaines, bien représentées en français, sont moins présentes en anglais.

Pour la suite de l'analyse, il est important de préciser que tous les titres, mots-clés et résumés ont été convertis en minuscules afin d'assurer des résultats homogènes et fiables.

## 6) Evaluation

Une évaluation manuelle consisterait à comparer les mots-clés générés à ceux d'une référence (Gold Standard). Bien que souvent efficace, cette méthode présente des inconvénients : elle est chronophage, énergivore et limite le traitement de gros volumes de données. En outre, elle peut introduire des biais en raison des perspectives et interprétations subjectives des évaluateurs humains.

Dans notre étude, nous avons opté pour une évaluation automatique, en comparant les mots-clés extraits automatiquement aux mots-clés de référence fournis par les auteurs sur HAL. Cette comparaison s'est effectuée en mode exact matching, où seuls les termes identiques étaient considérés comme des correspondances, afin d'assurer une évaluation précise et cohérente des résultats. Pour chaque article, les mots-clés les plus pertinents ont été extraits des résumés à l'aide de toutes les méthodes évaluées. Nous avons utilisé le F1-Score, une métrique couramment employée pour évaluer la performance des modèles d'extraction de mots-clés. Le F1-Score est la moyenne harmonique entre la précision, qui est le rapport entre le nombre de mots-clés correctement extraits et le nombre total de mots-clés extraits, et le rappel, qui mesure la proportion de mots-clés pertinents extraits par rapport au total des mots-clés pertinents dans le texte. Dans le contexte de l'extraction de mots-clés, un F1-Score élevé indique que le modèle parvient à extraire une proportion importante de mots-clés pertinents (haut rappel) tout en limitant l'extraction de mots-clés non pertinents (haute précision).

Pour chaque article et méthode d'extraction dans cette étude, un F1-Score est calculé. En moyennant ces scores sur l'ensemble du corpus, nous obtenons une évaluation comparative des performances des méthodes classiques et modernes<sup>1</sup>.

## 7) Résultats

Les résultats de l'évaluation sont présentés dans un tableau avec la précision, le rappel et le F1 score, classés par performance décroissante.

### 7-1 Les approches traditionnelles

Cette section présente les performances des modèles fondés sur des approches graphiques et statistiques, évalués selon les critères de précision, rappel et F1-Score. Les résultats sont également segmentés en fonction de l'inclusion ou non du titre dans l'évaluation.

Graph / Stat with title

Modèle	Precision	Recall	F1 Score
kw_by_pos_rank	0.062	0.115	0.080
kw_by_mp_rank	0.062	0.113	0.079
kw_by_topic_rank	0.059	0.108	0.076
kw_by_single_rank	0.053	0.098	0.068
kw_by_yake	0.053	0.098	0.068
kw_by_text_rank	0.039	0.072	0.050

Graph / Stat Without title

Modèle	Precision	Recall	F1 Score
kw_by_pos_rank	0.056	0.103	0.072
kw_by_mp_rank	0.056	0.103	0.072
kw_by_topic_rank	0.053	0.096	0.068
kw_by_yake	0.052	0.096	0.067
kw_by_single_rank	0.045	0.083	0.058
kw_by_text_rank	0.036	0.066	0.046

Figure 3: Résultats pour les modèles traditionnelles<sup>2</sup>

Ces résultats montrent une performance disparate, avec des écarts allant jusqu'à 60% entre les modèles les plus faibles (TextRank) et les plus performants (PositionRank, suivi de MultipartiteRank).

En ce qui concerne les modèles basés sur les embeddings, nous avons évalué le système KeyBERT. Dans ce système, il existe deux paramètres qui régissent la sélection des mots-clés: MMR (Maximal Marginal Relevance) et MSum (Maximum Similarity to Unselected). MMR équilibre la pertinence des mots-clés avec le texte source et leur diversité, en réduisant la redondance entre les termes sélectionnés. En revanche, MSum se concentre uniquement sur la maximisation de la similarité entre les mots-clés non encore sélectionnés et le texte source, favorisant une extraction plus pertinente mais potentiellement moins variée.

Keybert with title

Modèle	Precision	Recall	F1 Score
kw_by_keybert	0.058	0.081	0.067
kw_by_keybert_mmr_msum	0.052	0.073	0.061

Keybert without title

Modèle	Precision	Recall	F1 Score
kw_by_keybert	0.056	0.078	0.065
kw_by_keybert_mmr_msum	0.050	0.070	0.058

Figure 4: Résultats pour les modèles basés sur keyBERT

<sup>1</sup> Le code complet des évaluations est disponible ici: <https://github.com/obtic-sorbonne/keywords/tree/main>

<sup>2</sup> Les modèles utilisés incluent PositionRank (kw\_by\_pos\_rank), MultipartiteRank (kw\_by\_mp\_rank), TopicRank (kw\_by\_topic\_rank), SingleRank (kw\_by\_single\_rank), YAKE! (kw\_by\_yake) et TextRank (kw\_by\_text\_rank).

Les résultats pour KeyBERT selon ces variantes sont très similaires, avec des écarts minimes, ce qui indique des performances quasiment équivalentes entre ces méthodes. Nous observons également que l'inclusion du titre n'a pas d'effet significatif sur les résultats.

## 7-2 Les approches génératives

L'évaluation des approches basées sur les LLMs a été réalisée en testant des modèles génératifs open-source et des modèles propriétaires:

- Les modèles open-source sont: LLaMA-3 70B et LLaMA-3 8B ([Meta](#)); Mixtral 8\*7B et Mistral 7B ([Mistral](#)) ; Gemma 7B ([Google](#)).
- Les modèles propriétaires sont: GPT-3.5 Turbo et GPT-4o ([OpenAI](#)) ; Claude 3 Haiku et Claude 3 Instant ([Anthropic](#)).

LLM with title					LLM without title				
	Modèle	Precision	Recall	F1 Score		Modèle	Precision	Recall	F1 Score
	llama3-70b-8192	0.132	0.245	0.163		llama3-70b-8192	0.120	0.224	0.148
	claude-3-haiku-20240307	0.130	0.218	0.154		claude-3-haiku-20240307	0.120	0.204	0.143
	llama3-8b-8192	0.147	0.181	0.151		llama3-8b-8192	0.136	0.172	0.142
	gpt4o	0.075	0.222	0.108		gpt4o	0.071	0.206	0.101
	claude-instant-1.2	0.073	0.183	0.097		claude-instant-1.2	0.066	0.171	0.088
	kw_openai_gpt3.5	0.089	0.094	0.087		kw_openai_gpt3.5	0.086	0.089	0.083
	open-mixtral-8x7b	0.057	0.188	0.083		open-mistral-7b	0.047	0.176	0.070
	open-mistral-7b	0.050	0.199	0.077		open-mixtral-8x7b	0.048	0.156	0.069
	gemma-7b-it	0.051	0.079	0.059		gemma-7b-it	0.052	0.081	0.060

Figure 5: Résultats pour les modèles basé sur les LLMs

Les résultats pour les LLMs montrent une variabilité importante, avec des performances allant du simple au triple. L'inclusion des titres améliore les performances d'environ 10 %.

On observe que les trois meilleurs modèles sont LLaMA3 70B, Claude 3 Haiku, et LLaMA3 8B. En revanche, Gemma 7B affiche de faibles performances, principalement en raison du non-respect des prompts dans le format des résultats, ce qui pénalise fortement l'évaluation en exact matching.

## 8) Correspondance flexible: utilisation de la distance de Levenshtein

Jusqu'à présent, nous avons utilisé l'approche Exact Matching, une méthode de comparaison stricte des termes, sans tolérance pour les variations telles que les formes plurielles, l'usage des tirets ou éventuellement les erreurs typographiques. Cette approche, bien que précise, est rigide et réduit la capacité à capturer toutes les correspondances pertinentes.

Nous avons donc expérimenté le Fuzzy Matching, qui permet de comparer les mots-clés générés avec ceux de référence en prenant en compte les variations formelles. Plusieurs métriques peuvent attribuer un "score de proximité" entre deux chaînes de caractères, telles que Levenshtein, Jaro-Winkler, ainsi que différents modèles d'embeddings ([Alqahtani et al., 2021](#)). Dans cette étude, nous avons adopté la distance de Levenshtein, également appelée distance d'édition. Elle quantifie le nombre minimal d'opérations nécessaires pour transformer une chaîne de caractères en une autre,

les opérations possibles étant l'insertion, la suppression ou la substitution de caractères. Les résultats sont présentés sous forme de graphiques pour illustrer l'évolution du F1-Score à mesure que l'on augmente la flexibilité de la distance de Levenshtein (de 0 à 4).

## 8-1 Les approches traditionnelles

Ces modèles, bien qu'anciens, offrent une bonne robustesse dans des contextes où la complexité des termes est modérée et où la structure du texte joue un rôle important dans l'extraction des mots-clés.

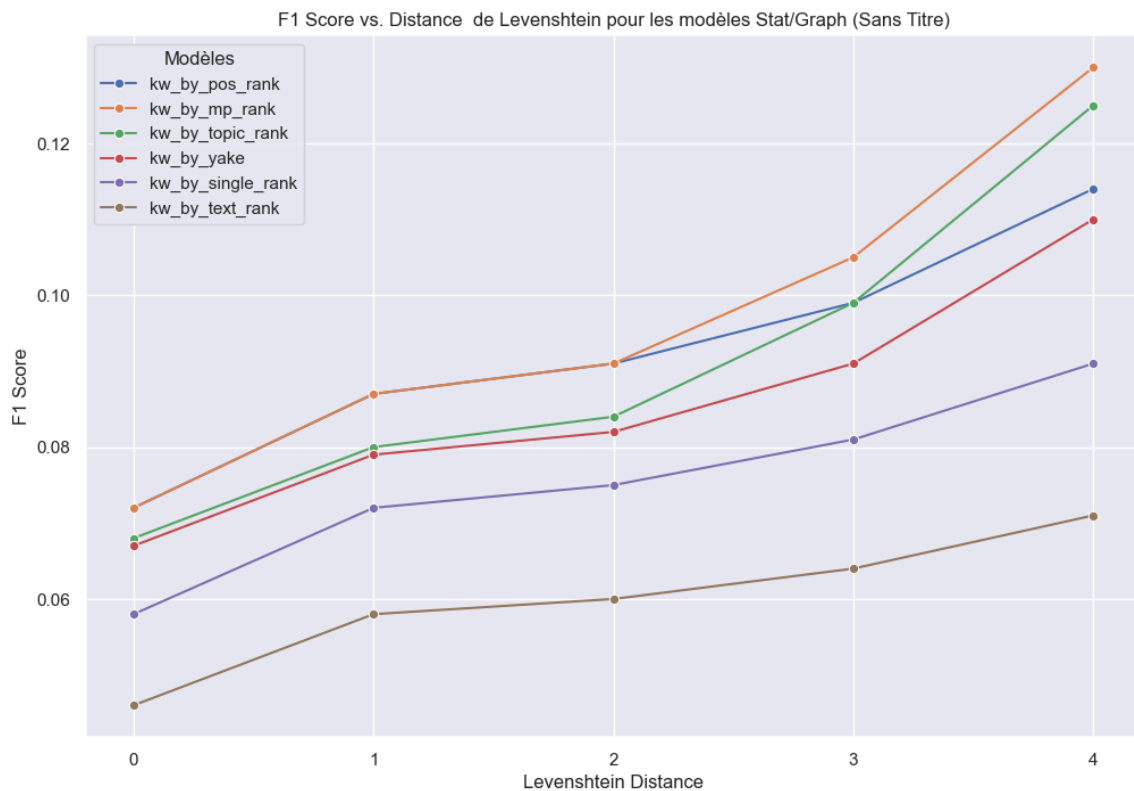


Figure 6 - Résultats pour les approches traditionnelles avec Levenshtein (**sans les titres**)

Les résultats confirment l'importance de la flexibilité offerte par la distance de Levenshtein pour mieux capturer les variations linguistiques.



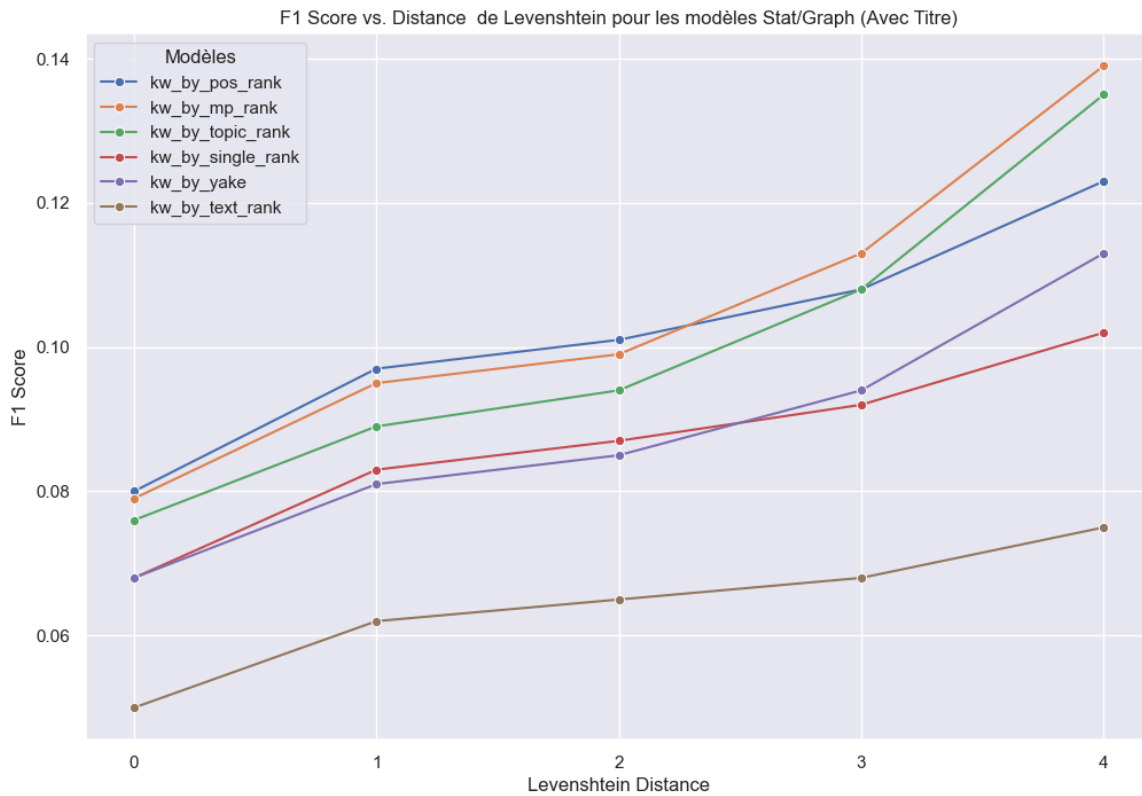


Figure 7 - Résultats pour les approches traditionnelles avec Levenshtein (**avec les titres**)

Il est à noter que l'ajout des titres améliore la performance des modèles traditionnelles, même si l'impact reste modéré.

KeyBERT tire parti des représentations contextuelles fournies par les modèles d'embeddings, offrant ainsi une approche plus nuancée pour l'extraction de mots-clés, notamment lorsque les termes sont répartis de manière moins homogène dans le texte.

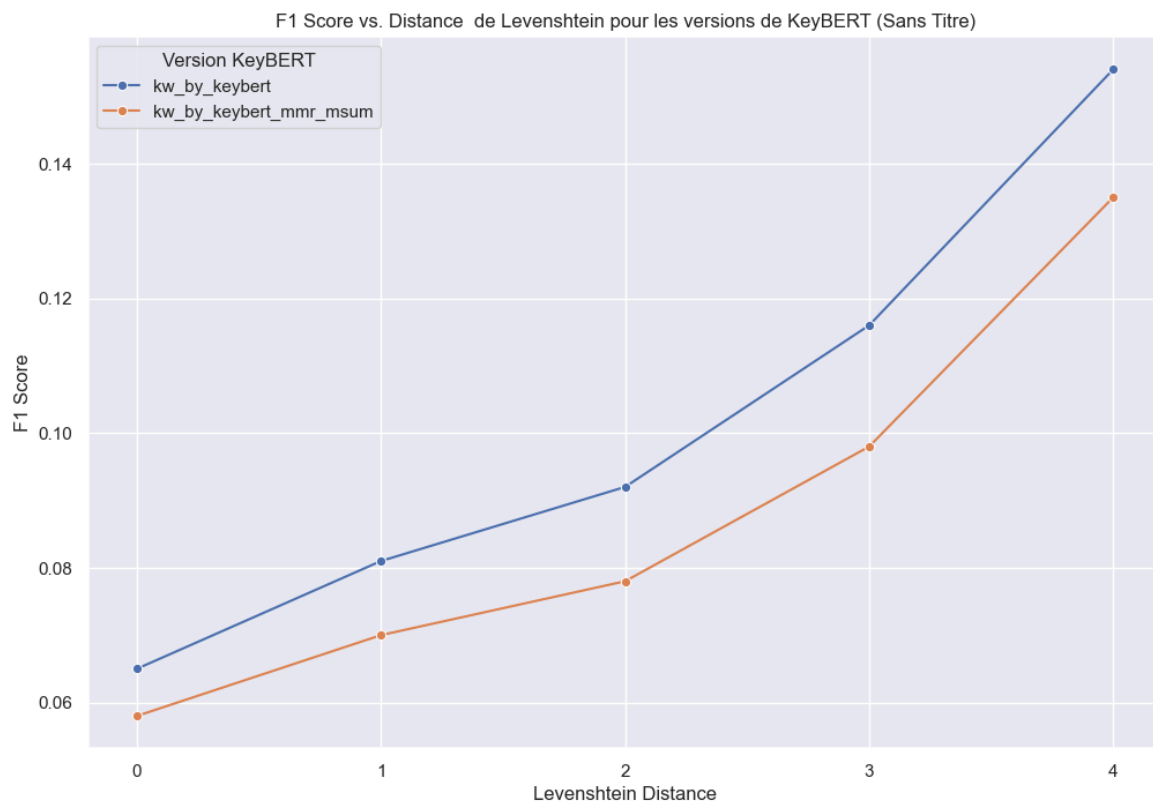


Figure 8 - Résultats pour keyBERT avec la distance de Levenshtein (**sans les titres**)

L'utilisation de la distance de Levenshtein dans ce contexte montre des gains intéressants en termes de précision, mais au détriment de la rapidité d'exécution.

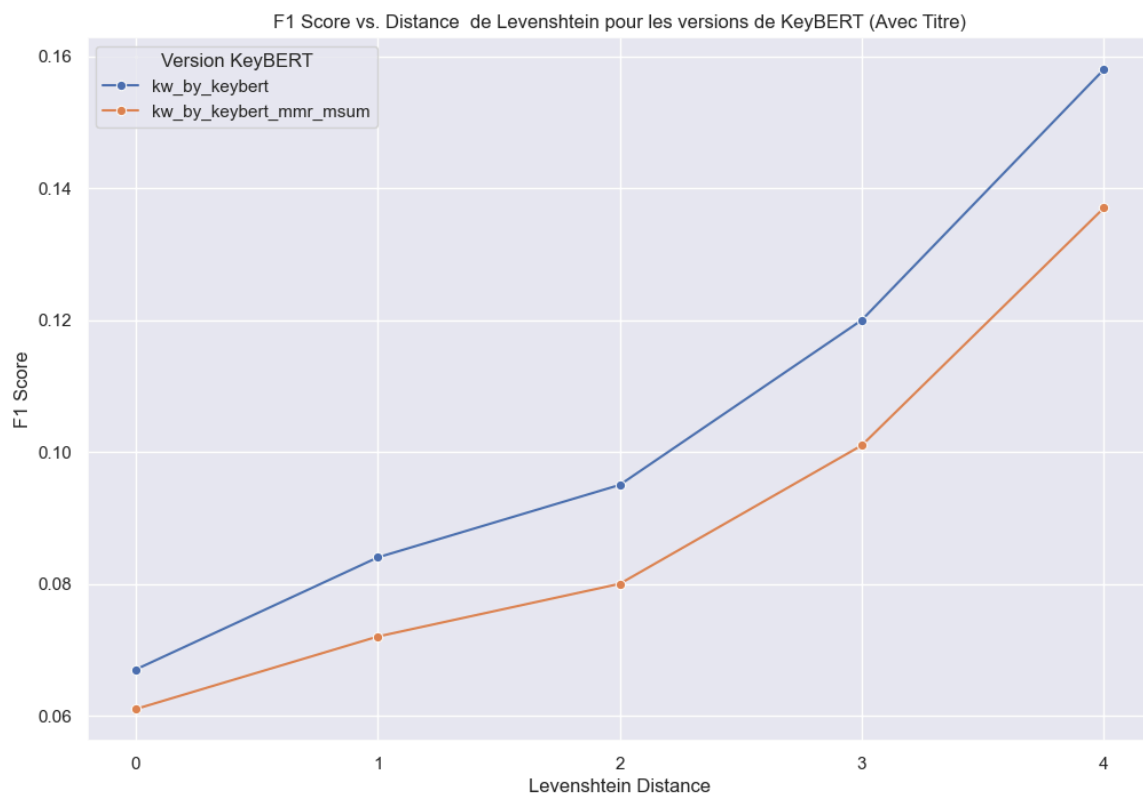


Figure 10 - Résultats pour keyBERT avec la distance de Levenshtein (**avec les titres**)

Avec les titres, on observe une légère augmentation de la performance, notamment pour les modèles basés sur KeyBERT.

## 8-2 Les approches génératives

Les modèles LLMs, grâce à leur compréhension approfondie du contexte et à leur capacité à traiter de grandes quantités de données, surpassent souvent les autres approches, notamment dans des tâches complexes ou pour des textes aux structures variées.

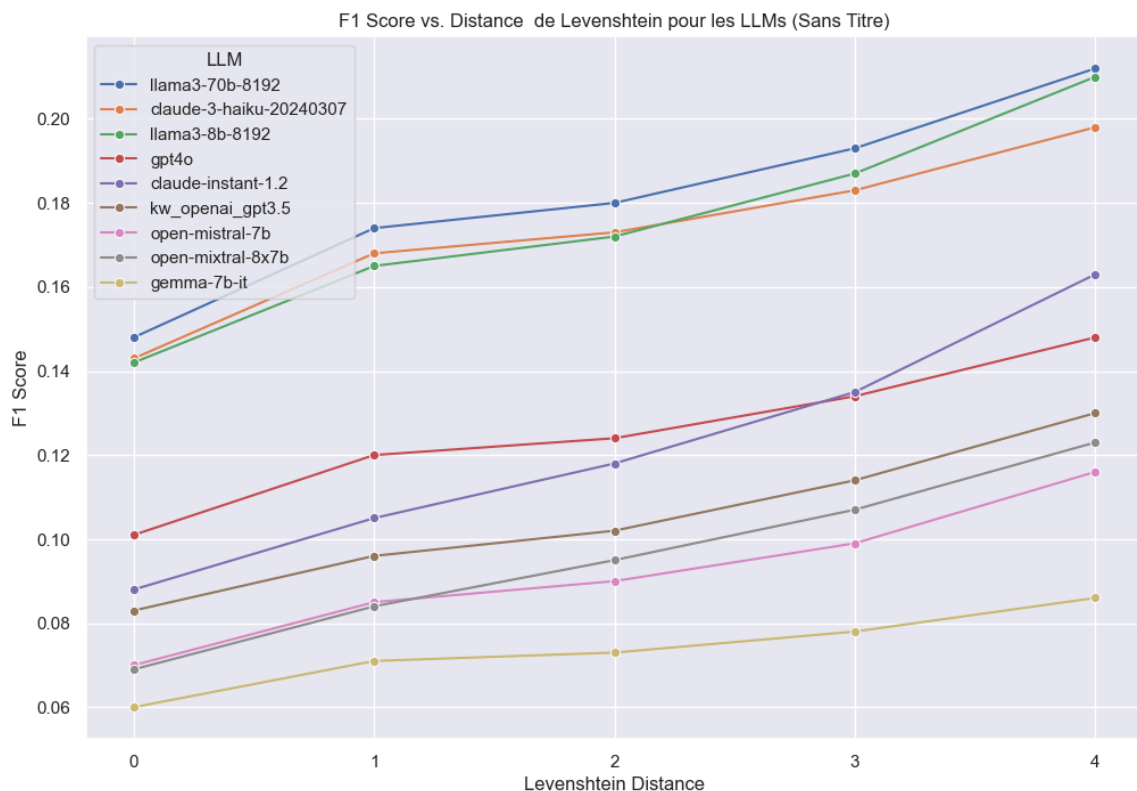


Figure 11 - Résultats pour les modèles génératifs avec la distance de Levenshtein (**sans les titres**)

Les modèles LLMs semblent bénéficier grandement de la souplesse offerte par la distance de Levenshtein.

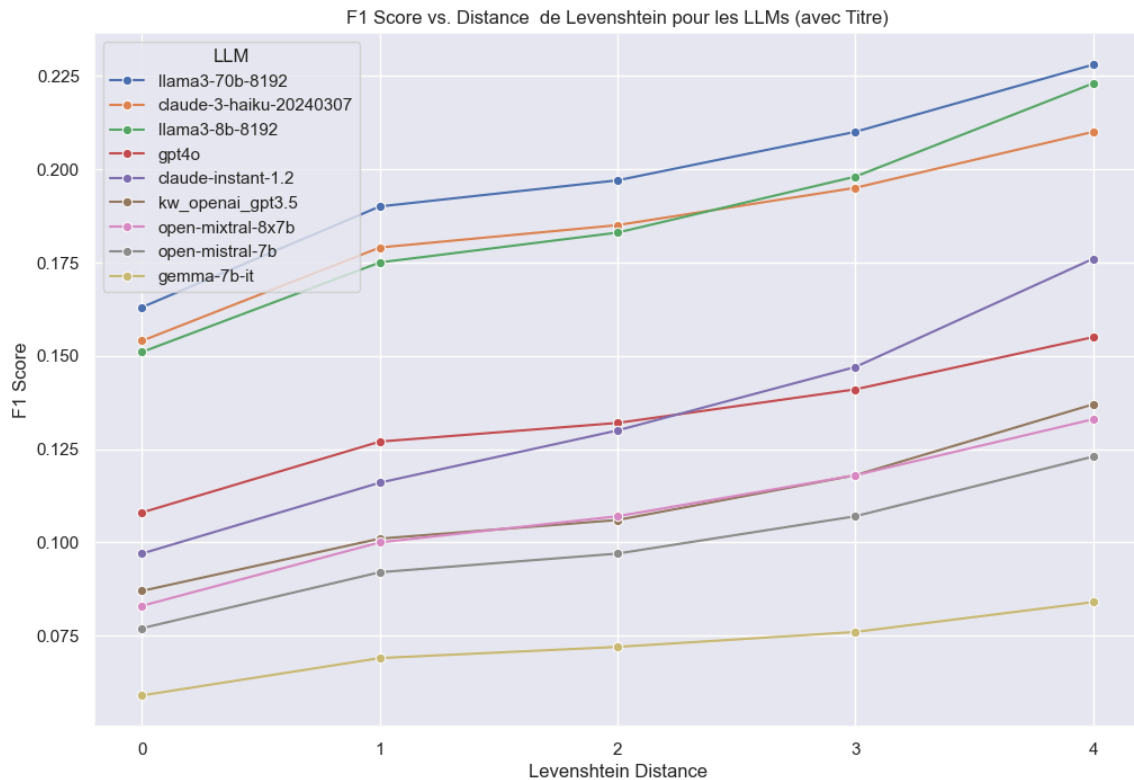


Figure 12 - Résultats pour les modèles génératifs avec la distance de Levenshtein (**avec les titres**)

L'inclusion des titres dans les modèles LLMs continue d'améliorer les scores, mais avec un retour décroissant au-delà d'un certain seuil.

## 9) LLMs et coût par token

L'utilisation de LLMs requiert des ressources matérielles importantes, ce qui peut limiter leur applicabilité pour des institutions avec des moyens techniques restreints. Cela pose également des enjeux environnementaux en raison de la consommation énergétique élevée. Le coût de traitement par token constitue donc un facteur déterminant. À F1 Score similaire, certains modèles peuvent être de 10 à 100 fois plus coûteux à exécuter. Dans le cadre de traitements de données massives, il devient essentiel d'intégrer cette dimension dans notre évaluation. Pour synthétiser cette approche, nous introduisons une nouvelle métrique dédiée aux LLMs qui combine la performance (F1 Score) et le coût (\$/million de tokens) : le Token Efficiency Score (TES). Cette métrique intègre les performances et les coûts, en appliquant une pénalisation plus forte sur les coûts, tout en privilégiant la performance. Nous utilisons une moyenne harmonique pondérée, définie par la formule suivante (avec  $\alpha = 10$ ) :

$$\text{TES} = \frac{(1 + \alpha) \times F_1 \times \text{Prix}}{\alpha \times \text{Prix} + F_1}$$

Le calcul montre que les modèles les plus performants sont également parmi les moins coûteux, notamment Llama-3 70B, Llama-3 8B, et Claude 3 Haiku.

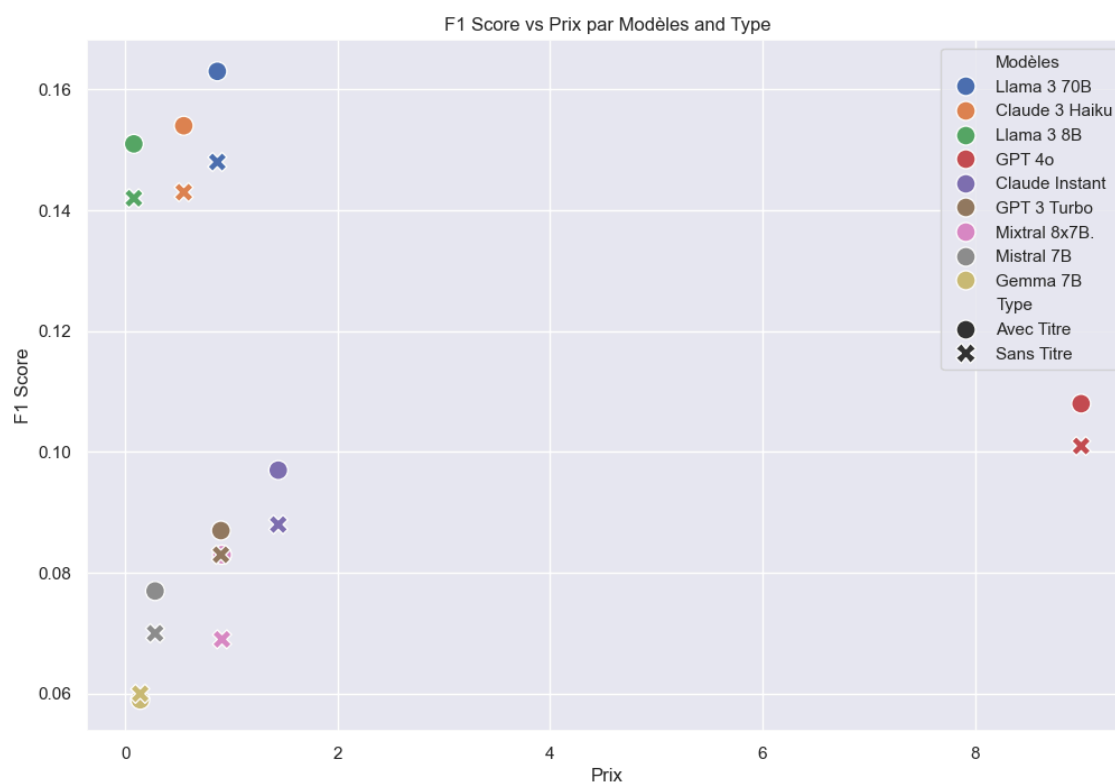


Figure 13 - Résultats pour les LLMs selon le coût pour le dataset entier (en euro)

Dans la figure 14 qui suit, on ordonne les modèles génératifs selon leur score TES du plus efficient au moins efficient. On retrouve comme prévu les trois modèles en tête : Llama 3 70B - Claude 3 Haiku - Llama 3 8B en tête et Gemma 7B de Google en dernière position.

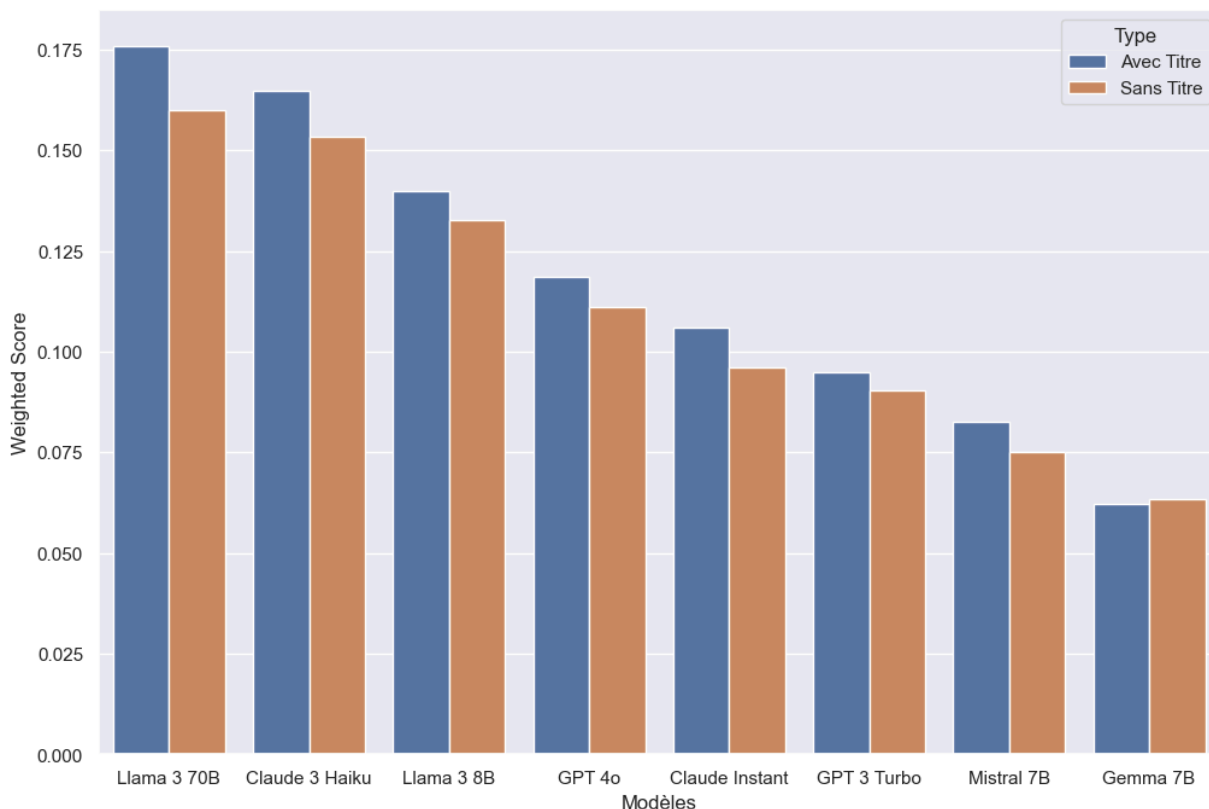


Figure 14 - Résultats TES pour les modèles basés sur les LLMs

Le TES permet d'identifier clairement les modèles les plus performants tout en prenant en compte le facteur coût, crucial dans des scénarios à grande échelle.

## 10) Limites

Les LLMs apportent des avancées significatives dans l'extraction de mots-clés, mais leur utilisation présente certaines limites. Tout d'abord, leur formation sur des corpus génériques limite leur précision dans des domaines scientifiques spécifiques, ce qui nécessite souvent un fine-tuning sur des données annotées pour capturer les termes techniques propres à chaque discipline. De plus, les LLMs fonctionnent comme des "boîtes noires" difficiles à interpréter, rendant complexe l'explication des choix de mots-clés générés et limitant ainsi leur adoption dans des contextes exigeant une traçabilité des méthodes.

La sensibilité des LLMs aux prompts et leur tendance à produire des réponses variables introduisent également une certaine instabilité dans les résultats (*stochasticity*). Cela impose des ajustements fréquents des prompts pour obtenir des mots-clés cohérents, et oblige à répéter les évaluations pour lisser les fluctuations des F1-Scores. Par ailleurs, l'utilisation de prompts détaillés et conversationnels peut alourdir les coûts de traitement sans garantir une amélioration de la pertinence des mots-clés extraits, bien que cela permette d'affiner la structure des mots-clés générés.

Enfin, le corpus HAL lui-même comporte des limites pour l'extraction de mots-clés. De nombreux mots-clés fournis par les auteurs ne figurent ni dans les résumés ni dans les titres des articles, ce qui pose un défi pour les modèles extractifs et introduit un biais lors de l'évaluation de leur performance.

## 11) Conclusion et perspectives

Les résultats de la comparaison montrent que les approches génératives basées sur les LLMs surpassent les méthodes traditionnelles en termes de précision et de pertinence des mots-clés extraits. Ces modèles modernes, même en configuration zero-shot learning, parviennent à capturer des nuances contextuelles et des relations sémantiques que les méthodes statistiques traditionnelles ne peuvent pas toujours saisir, rendant l'extraction de mots-clés plus riche et plus représentative des contenus scientifiques. L'introduction de la nouvelle métrique TES, que nous avons conçue, a révélé que les modèles offrant le meilleur compromis entre qualité et coût sont également les moins onéreux, confirmant leur efficacité économique. De plus, l'intégration des titres, riches en information, améliore significativement les scores F1 sans augmenter notablement la charge en tokens.

Plusieurs pistes restent à explorer pour améliorer l'extraction automatique de mots-clés. En priorité, le calibrage des prompts pourrait affiner les résultats en fournissant aux LLMs des indications spécifiques, comme le nombre ou la longueur des mots-clés souhaités, tout en veillant à ne pas alourdir les requêtes API. De plus, structurer les prompts sous forme d'une conversation simulée entre l'utilisateur et le modèle pourrait améliorer la consistance des réponses en réduisant les variations dans le format des mots-clés générés, avec un impact direct sur le F1-Score, en particulier pour des modèles comme Gemma de Google.

Une autre piste prometteuse est de fine-tuner un modèle LLM, ce qui permettrait d'intégrer des approches modernes tout en améliorant les performances des méthodes extractives. À moyen terme, élargir le traitement aux textes complets des articles, plutôt qu'aux seuls résumés, offrirait des opportunités pour extraire des mots-clés encore plus représentatifs du contenu global, notamment dans des contextes scientifiques (Teufel et Moens, 2002).

Pour aller plus loin, l'utilisation de nouvelles métriques, comme le NPMI, la cohérence de sujet ou BM25, pourrait fournir des outils complémentaires pour évaluer la pertinence et la consistance des mots-clés générés.

Enfin, une modélisation thématique à l'aide d'outils comme BERTopic pourrait dépasser la simple extraction de mots-clés pour regrouper les résultats en thèmes structurés, améliorant ainsi la compréhension et l'organisation contextuelle des contenus.

+ Seen with Hamed

## Bibliographie

Alqahtani, A., Alhakami, H., Alsubait, T., & Baz, A. (2021). A Survey of Text Matching Techniques. *Engineering, Technology & Applied Science Research*, 11(1), 6656–6661.

Beliga, S., Meštrović, A., & Martinčić-Ipšić, S. (2015). An Overview of Graph-Based Keyword Extraction Methods and Approaches. *Journal of Information and Organizational Sciences*, 39(1), 1–20.

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., & Jaggi, M. (2018). Simple Unsupervised Keyphrase Extraction Using Sentence Embeddings. *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, 221–229. <https://doi.org/10.18653/v1/K18-1022>

Boudin, F. (2018). Unsupervised Keyphrase Extraction with Multipartite Graphs. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 667–672. New Orleans, LA: Association for Computational Linguistics.

Bougouin, A., Boudin, F., & Daille, B. (2013). TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. *International Joint Conference on Natural Language Processing (IJCNLP)*, 543–551. Nagoya, Japan.

Boukhaled, M., & Lefèvre, F. (2022). A Survey on Keyphrase Extraction in Natural Language Processing. *Journal of Artificial Intelligence Research*, 73, 1–45. <https://doi.org/10.1613/jair.1.13398>

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword Extraction from Single Documents Using Multiple Local Features. *Information Sciences*, 509, 257–289.

Church, K. W., & Gale, W. A. (1995). Poisson Mixtures. *Natural Language Engineering*, 1(2), 163–190. <https://doi.org/10.1017/S1351324900000135>

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>

El-Beltagy, S. R., & Rafea, A. (2009). KP-Miner: A Keyphrase Extraction System for English and Arabic Documents. *Information Systems*, 34(1), 132–144.

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

Grootendorst, M. (2020). BERTopic: Leveraging BERT and c-TF-IDF to Create Easily Interpretable Topics. <https://github.com/MaartenGr/BERTopic>

Justeson, J. S., & Katz, S. M. (1995). Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text. *Natural Language Engineering*, 1(1), 9–27. <https://doi.org/10.1017/S1351324900000048>

Kulumba, F., Antoun, W., Vimont, G., & Romary, L. (2024). Harvesting Textual and Structured Data from the HAL Publication Repository. *arXiv preprint*, arXiv:2407.20595v1. <https://doi.org/10.48550/arXiv.2407.20595>

Mihalcea, R., and P. Tarau. 2004. "TextRank: Bringing Order into Text." In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. Barcelona, Spain: Association for Computational Linguistics.

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. <https://arxiv.org/abs/1301.3781>

Pagliardini, M., Gupta, P., & Jaggi, M. (2017). Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. <https://arxiv.org/abs/1703.02507>

Papagiannopoulou, E., & Tsoumakas, G. (2020). A Review of Keyphrase Extraction. *WIREs Data Mining and Knowledge Discovery*, 10, e1339. <https://doi.org/10.1002/widm.1339>



Salton, G., & Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)

Schopf, T., Klimek, S., & Matthes, F. (2023). PatternRank: Leveraging Pretrained Language Models and Part of Speech for Unsupervised Keyphrase Extraction. *Department of Computer Science, Technical University of Munich*.

Song, M., Feng, Y., & Jing, L. (2023). A Survey on Recent Advances in Keyphrase Extraction from Pre-trained Language Models. *Findings of the Association for Computational Linguistics: EACL 2023*, 2153–2164. Dubrovnik, Croatia: Association for Computational Linguistics.

Turney, P. D. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2(4), 303–336. <https://doi.org/10.1023/A:1009976629275>

Vaswani, A., Shard, P., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.

+

- Teufel, S., & Moens, M. (2002). **Summarizing scientific articles: Experiments with relevance and rhetorical status**. *Computational Linguistics*, 28(4), 409–445.  
DOI: 10.1162/089120102762671936

# Extraction de mots-clés à partir de publications scientifiques : l'apport des LLMs

Motasem ALRAHABI, Nacef BEN MANSOUR  
Sorbonne Université  
prenom-nom@sorbonne-universite.fr

## **Résumé:**

L'extraction automatique des mots-clés est cruciale pour résumer le contenu des documents et optimiser la recherche d'informations. Dans cette étude, nous comparons les performances de plusieurs modèles d'extraction et de génération de mots-clés appliqués aux résumés d'articles issus des archives HAL : des approches basées sur des statistiques et des modèles vectoriels, ainsi que des approches génératives modernes utilisant les LLMs. Les résultats montrent que les LLMs surpassent largement les méthodes traditionnelles en termes de précision et de pertinence, même en configuration zero-shot, et que l'inclusion des titres améliore significativement les scores F1. Nous introduisons également une nouvelle métrique pour évaluer les performances des LLMs en tenant compte des coûts de traitement, offrant ainsi une perspective équilibrée entre efficacité et coût.

## **Mots-clés:**

Extraction de mots-clés, modèles de langage, évaluation, corpus HAL.

## **Contexte :**

Ce travail s'inscrit dans un nouveau projet visant à recenser les compétences et l'expertise des laboratoires et des chercheurs de Sorbonne Université. En s'appuyant sur les publications scientifiques, l'objectif est de développer un système intelligent permettant d'interroger et de recommander des informations liées à l'expertise des chercheurs, afin de faciliter les collaborations et d'accroître leur visibilité scientifique.

## **Introduction:**

L'extraction de mots-clés est essentielle pour identifier les termes pertinents d'un document ou corpus, facilitant l'analyse et l'exploitation des données. Elle s'applique notamment à la condensation de contenu, le résumé automatique et la détection de tendances. Les méthodes traditionnelles, souvent limitées dans la prise en compte du contexte et de la sémantique, ont évolué grâce aux modèles d'apprentissage profond, qui combinent analyse statistique et compréhension contextuelle, comme illustré par Song et al. (2023).

## **Approches pour l'extraction de mots-clés:**

L'évolution des approches pour l'extraction de mots-clés reflète les progrès en traitement du langage naturel:

- Règles linguistiques : Basées sur des syntagmes nominaux et n-grammes (ex. YAKE! de Campos et al., 2020), elles exploitent des caractéristiques syntaxiques pour extraire des mots-clés.
- Approches statistiques : Méthodes comme TF-IDF (Salton & Buckley, 1988) ou RAKE identifient les mots-clés selon leur fréquence et co-occurrence. Les distributions statistiques complexes ont aussi été explorées (Church & Gale, 1995).
- Apprentissage supervisé : Entraînées sur des données annotées, ces méthodes incluent KP-Miner et des modèles de classification avancés (Papagiannopoulou et al., 2020).
- Apprentissage non supervisé : Utilisant des graphes, des méthodes comme TextRank (Mihalcea & Tarau, 2004) ou TopicRank (Bougouin et al., 2013) classent les mots selon leur importance contextuelle.

- Vectorisation : Des embeddings vectoriels enrichis (ex. EmbedRank basé sur Word2Vec ou Sent2Vec) et des modèles comme KeyBERT intègrent le contexte sémantique (Bennani-Smires et al., 2018; Grootendorst, 2020).
- Approches génératives : Les LLMs (BERT, GPT, LLaMA) exploitent les Transformers pour capturer des relations complexes, permettant une extraction zero-shot ou fine-tunée et générant des mots-clés reformulés (Vaswani et al., 2017).

### **Méthodologie:**

- Préparation des données: Extraction des résumés et mots-clés d'auteurs depuis HAL, avec deux configurations : résumés + titres et résumés seuls.
- Mise en œuvre des approches : Comparaison des méthodes génératives modernes (LLMs en zero-shot) et traditionnelles.
- Évaluation : Analyse de la pertinence et précision des mots-clés selon la précision, le rappel et la mesure F1.
- Analyse des résultats : Comparaison statistique et qualitative des performances des deux catégories de méthodes.

Cette méthodologie garantit une comparaison robuste entre approches.

### **Les données :**

Les données utilisées sont celles des chercheurs de Sorbonne Université, et proviennent de la plateforme HAL, une archive ouverte valorisée par des travaux récents comme HALvest (Kulumba et al., 2024). Le corpus, constitué via l'API HAL, contient 4 700 articles exploitables, soit 30 % des 12 000 initialement collectés après élimination des doublons et des articles sans mots-clés ou résumés. Les articles sélectionnés, issus de chercheurs affiliés au Sorbonne Center for Artificial Intelligence (SCAI), couvrent divers domaines scientifiques. La majorité des articles (85 %) est en anglais, avec une moyenne de 5,35 mots-clés par article et une longueur moyenne de 2,14 mots. L'informatique domine dans les deux langues, suivi des sciences humaines en français et des sciences de la vie en anglais. Les titres, mots-clés et résumés ont été uniformisés en minuscules pour garantir des résultats fiables.

### **Evaluation :**

L'évaluation de cette étude repose sur une approche automatique comparant les mots-clés extraits à ceux fournis par les auteurs sur HAL, en utilisant une correspondance exacte (*exact matching*) et une correspondance approximative (*fuzzy matching*). Les performances des méthodes sont mesurées via le F1-Score, une métrique combinant précision et rappel.

### **Résultats avec correspondance exacte**

1) Approches traditionnelles : Les performances des approches traditionnelles révèlent une variabilité importante, avec des écarts allant jusqu'à 60% entre les modèles les plus faibles (TextRank) et les plus performants (PositionRank, suivi de MultipartiteRank).

Quant aux modèles basés sur des embeddings, KeyBERT a été testé avec deux paramètres, MMR et MSum. MMR favorise la diversité en réduisant la redondance, tandis que MSum optimise la pertinence en maximisant la similarité. Les résultats pour KeyBERT montrent des performances quasi équivalentes entre ces variantes, et l'inclusion ou non des titres n'a pas d'impact notable sur les scores.

2) Approches génératives : L'évaluation des approches basées sur les LLMs a comparé des modèles génératifs open-source (LLaMA-3, Mixtral, Gemma) et propriétaires (GPT-3.5, GPT-4, Claude 3). Les performances varient considérablement, avec un gain d'environ 10 % lorsque les titres sont inclus. Les meilleurs résultats sont obtenus par LLaMA-3 70B, Claude 3 Haiku, et LLaMA-3 8B, tandis que Gemma 7B affiche des performances faibles, pénalisées par un formatage incorrect des résultats impactant l'exact matching.

### **Résultats avec correspondance approximative**

Nous avons appliqué un Fuzzy Matching basé sur la distance de Levenshtein, permettant de capturer les variations formelles des mots-clés. Les résultats illustrent l'évolution du F1-Score à mesure que l'on augmente la flexibilité de la distance de Levenshtein (de 0 à 4).

#### **1) Approches traditionnelles :**

Les résultats confirment l'importance de la flexibilité offerte par la distance de Levenshtein pour mieux capturer les variations linguistiques.

KeyBERT tire parti des représentations contextuelles fournies par les modèles d'embeddings, offrant ainsi une approche plus nuancée pour l'extraction de mots-clés, notamment lorsque les termes sont répartis de manière moins homogène dans le texte.

Avec les titres, on observe une légère augmentation de la performance, notamment pour les modèles basés sur KeyBERT.

#### **2) Approches génératives :**

Les modèles LLMs semblent bénéficier grandement de la souplesse offerte par la distance de Levenshtein. L'inclusion des titres continue d'améliorer les scores, mais avec un retour décroissant au-delà d'un certain seuil.

### **LLMs et coût par token :**

L'utilisation des LLMs, bien que performante, est limitée par des contraintes matérielles, des coûts énergétiques élevés et des frais de traitement par token, pouvant varier considérablement. Afin d'évaluer leur efficacité dans un contexte de données importantes, nous avons introduit une nouvelle métrique, le Token Efficiency Score (TES), combinant le F1-Score et le coût (\$/million de tokens) via une moyenne harmonique pondérée. La TES pénalise davantage les coûts tout en valorisant la performance, elle est définie par la formule suivante (avec  $\alpha = 10$ ) :

Le calcul montre que les modèles les plus performants sont également parmi les moins coûteux, notamment Llama-3 70B, Llama-3 8B, et Claude 3 Haiku.

### **Conclusion et discussion :**

Notre étude a permis de constater que les LLMs surpassent les méthodes classiques en termes de précision et de pertinence, même en mode zero-shot. De plus, l'introduction de la formule TES révèle que les modèles les moins coûteux offrent le meilleur compromis entre qualité et coût. Par ailleurs, l'intégration des titres permet d'améliorer les scores F1 sans surcharger le nombre de tokens.

Les principales limites identifiées concernent, d'une part, le fonctionnement des LLMs, souvent perçus comme des "boîtes noires", ce qui complique leur interprétabilité, et d'autre part leur sensibilité aux prompts, qui génère des résultats instables nécessitant des ajustements fréquents. Par ailleurs, l'usage de prompts détaillés augmente les coûts sans garantir d'améliorations significatives. En ce qui concerne les données HAL, de nombreux mots-clés fournis par les auteurs sont absents des résumés ou des titres, ce qui complique l'évaluation des modèles extractifs et introduit un biais.

Nous envisageons d'approfondir l'analyse de certaines sections des publications (l'introduction ou la conclusion) afin d'extraire des mots-clés plus riches et pertinents (Teufel et Moens, 2002). Nous prévoyons également d'optimiser les prompts pour améliorer la précision sans alourdir la complexité des requêtes. Enfin, nous projetons de fine-tuner un LLM sur un nouveau "gold standard" afin d'en renforcer les performances. Le code associé à cette étude est disponible à l'adresse suivante : <https://github.com/obtic-sorbonne/keywords>.

### **Bibliographie :**

- Song, M. et al. (2023). Advances in Keyphrase Extraction from LLMs. Findings of ACL: EACL 2023, 2153–2164.
- Salton, G. & Buckley, C. (1988). Term-weighting in Text Retrieval. IP&M, 24(5), 513–523.
- Church, K. W. & Gale, W. A. (1995). Poisson Mixtures. Nat. Lang. Eng., 1(2), 163–190.








- Papagiannopoulou, E. & Tsoumakas, G. (2020). Keyphrase Extraction Review. *WIREs Data Min.*, 10, e1339.
- Mihalcea, R. & Tarau, P. (2004). TextRank: Ordering Text. *EMNLP*, 404–411.
- Bougouin, A. et al. (2013). TopicRank for Keyphrase Extraction. *IJCNLP*, 543–551.
- Bennani-Smires, K. et al. (2018). Unsupervised Keyphrase Extraction. *CoNLL*, 221–229.
- Grootendorst, M. (2020). BERTopic. [GitHub](<https://github.com/MaartenGr/BERTopic>).
- Vaswani, A. et al. (2017). Attention is All You Need. *NeurIPS*, 30, 5998–6008.
- Kulumba, F. et al. (2024). Data from HAL Repository. *arXiv:2407.20595*.
- Teufel, S. & Moens, M. (2002). Summarizing Scientific Articles. *Comp. Linguistics*, 28(4), 409–445.

# Annexes

## Autres références (ne pas supprimer):

<https://hal.science/hal-02643329/document>  
<https://www.oncrawl.com/technical-seo/automatically-extract-concepts-keywords-from-text-traditional-methods/>  
<https://encyclopedia.pub/entry/51973>  
[https://scholar.google.fr/scholar?as\\_ylo=2020&q=keyword+extraction+evaluation&hl=en&as\\_sdt=0,5&as\\_vis=1](https://scholar.google.fr/scholar?as_ylo=2020&q=keyword+extraction+evaluation&hl=en&as_sdt=0,5&as_vis=1)  
[A survey and classification of semantic search approaches Christoph Mangold](#)  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9753895/>  
<https://github.com/MaartenGr/KeyBERT>  
<https://www.kaggle.com/code/akhatova/extract-keywords>  
<https://spotintelligence.com/2022/12/13/keyword-extraction/>  
<https://towardsdatascience.com/keyword-extraction-methods-the-overview-35557350f8bb>  
<https://paperswithcode.com/task/keyword-extraction/codeless>  
<https://www.linkedin.com/pulse/keyword-extraction-lakebrains-technologies/>  
<https://medium.com/@gil.fernandes/keyword-extraction-with-langchain-and-chatgpt-770c04d378b4>  
<https://www.johnsnowlabs.com/the-experts-guide-to-keyword-extraction-from-texts-with-spark-nlp-and-python/>  
<https://fr.oncrawl.com/seo-technique/extraire-automatiquement-concepts-mots-cles-texte-methode-s-classiques/>  
<https://fr.oncrawl.com/seo-technique/extraire-automatiquement-concepts-mots-cles-texte-part-ii-approche-semantic/>  
<https://hal.science/hal-00821671/document>  
<https://www.grafiati.com/fr/literature-selections/extraction-de-connaissances-de-donnees/dissertation/>  
<https://www.seoquantum.com/billet/extraction-mots-cles#:~:text=L'extraction%20de%20mots%20%C3%A9s%20>  
<https://nlpccloud.com/fr/nlp-keyword-keyphrase-extraction-gpt-j-api.html>  
<https://seoarmy.fr/extraction-mots-cles/>  
<https://www.bew-web-agency.fr/quappelle-t-on-extraction-mots-cles/>  
<https://products.aspose.app/html/fr/keywords-extractor>  
<https://semji.com/fr/blog/les-meilleurs-outils-gratuits-pour-travailler-les-mots-cles-de-google-suggest/>

## Autres liens:

-  [stage Waly](#)  [Stage Waly \(présentation\)](#)  [Stage Waly \(notebook\).ipynb](#)
-  [Stage Keywords Ontologies - Xavier](#)
- [Projet Expertise SU:](#)  [Expertise: synthèse.docx](#)
-  [Similarité](#)
-  [Keywords Présentation](#)
- <https://aclanthology.org/2023.findings-eacl.161.pdf>

- <https://medium.com/@kanerika/generative-vs-discriminative-understanding-machine-learning-models-87e3d2b3b99f>
- <https://link.springer.com/article/10.1007/s42979-022-01481-7>
- <https://arxiv.org/pdf/2101.09990>
- <https://arxiv.org/pdf/2310.09151>
- [https://langnet.uniri.hr/papers/beliga/Beliga\\_KeywordExtraction\\_a\\_review\\_of\\_methods\\_and\\_approaches.pdf](https://langnet.uniri.hr/papers/beliga/Beliga_KeywordExtraction_a_review_of_methods_and_approaches.pdf)
- [https://www.researchgate.net/profile/Sifatullah-Siddiqi/publication/272372039\\_Keyword\\_and\\_Keyphrase\\_Extraction\\_Techniques\\_A\\_Literature\\_Review/links/58ac7db0aca272af0666243f/Keyword-and-Keyphrase-Extraction-Techniques-A-Literature-Review.pdf](https://www.researchgate.net/profile/Sifatullah-Siddiqi/publication/272372039_Keyword_and_Keyphrase_Extraction_Techniques_A_Literature_Review/links/58ac7db0aca272af0666243f/Keyword-and-Keyphrase-Extraction-Techniques-A-Literature-Review.pdf)
-