How Well Do Large Language Models Extract Keywords? A Systematic Evaluation on Scientific Corpora

Nacef Ben Mansour
Sorbonne Université
Paris, France

Hamed Rahimi
ISIR, Sorbonne Université
Paris, France

Motasem Alrahabi*
ObTIC, Sorbonne Université
Paris, France

Abstract

Automatic keyword extraction from scientific articles is pivotal for organizing scholarly archives, powering semantic search engines, and mapping interdisciplinary research trends. However, existing methods—including statistical and graph-based approaches—struggle to handle domain-specific challenges such as technical terminology, cross-disciplinary ambiguity, and dynamic scientific jargon. This paper presents an empirical comparison of traditional keyword extraction methods (e.g. TextRank and YAKE) with approaches based on Large Language Model. We introduce a novel evaluation framework that combines fuzzy semantic matching based on Levenshtein Distance with exact-match metrics (F1, precision, recall) to address inconsistencies in keyword normalization across scientific corpora. Through an extensive ablation study across nine different LLMs, we analyze their performance and associated costs. Our findings reveal that LLM-based methods consistently achieve superior precision and relevance compared to traditional approaches. This performance advantage suggests significant potential for improving scientific search systems and information retrieval in academic contexts.

1 Introduction

Keyword extraction algorithms are a group of statistical techniques that aim to identify the most relevant and representative terms for documents (Firoozeh et al., 2020). These methods have a wide range of applications, from improving information retrieval (Bracewell et al., 2005) and search engine optimization (Horasan, 2021) to information extraction, automatic document summarization (Bharti and Babu, 2017), and emerging trend detection (Kim et al., 2015). Over the years, the methodologies for keyword extraction have evolved significantly, reflecting advances in both linguistic understanding and computational techniques.

Traditional approaches, such as YAKE! (Campos et al., 2020), utilized syntactic analyses like noun or n-gram phrases to extract linguistic characteristics, including factors such as word position and frequency. Statistical techniques, including TF-IDF (Salton and Buckley, 1990) and RAKE (Rose et al., 2010), introduced quantitative measures to assess the importance of terms within a text and across corpora. While early methods primarily relied on linguistic rules and statistical measures, recent advancements have embraced deep learning to capture both contextual and semantic nuances. This shift has been driven by the emergence of large language models (LLMs) (Song et al., 2023a), which leverage the Transformer architecture (Vaswani, 2017) to understand and generate text with remarkable contextual depth. LLMs excel at modeling complex relationships within text, enabling precise keyword extraction through zero-shot, few-shot, or fine-tuned approaches. Unlike traditional extractive methods, which are confined to selecting explicit terms from the text, generative models can create or rephrase keywords that encapsulate the underlying meaning, even when such terms are absent in the original text. In (Song et al., 2023b), the authors evaluate the performance of ChatGPT and ChatGLM in extracting keyphrases without prior fine-tuning, highlighting their effectiveness in identifying relevant terms. Meanwhile, (Maragheh et al., 2023) explores a multi-stage approach to keyword extraction in an e-commerce setting, aiming to refine results by filtering out non-informative or sensitive keywords and mitigating hallucinations. In this work, we present a comprehensive analysis of keyword extraction methods by bridging traditional approaches and LLMs. Specifically, we conduct a comparative evaluation of these methodologies, examining their strengths, limitations, and practical applications. Our study employs two matching

*motasem.alrahabi@sorbonne-universite.fr

Proceedings of the 1st Workshop on AI and Scientific Discovery: Directions and Opportunities, pages 13–21

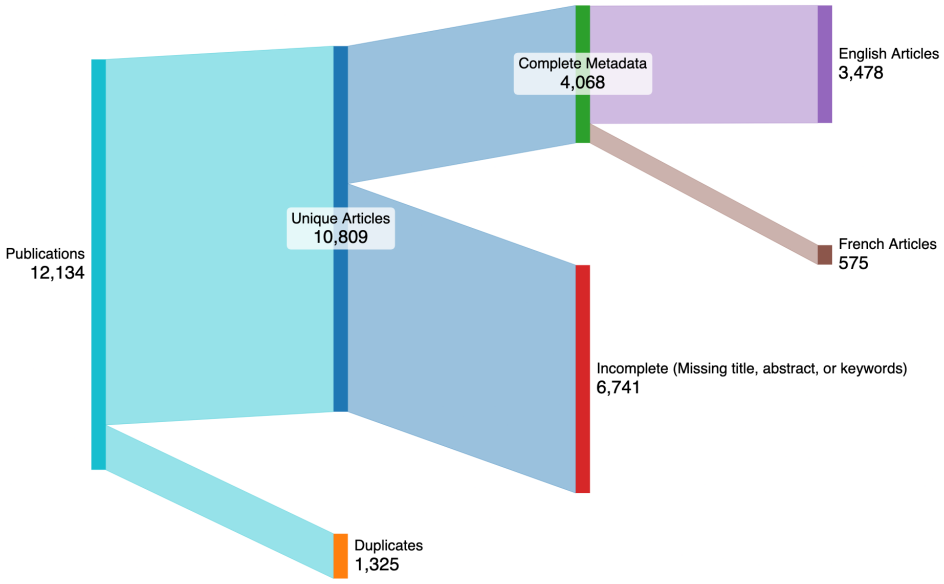


Figure 1: **Distribution of articles by language in HAL**

techniques—exact matching and flexible matching—to assess the effectiveness of keyword extraction. Furthermore, we perform an ablation study to investigate the performance and computational cost of different LLMs, providing insights into their trade-offs and suitability for various scenarios.

2 Related Works

The evolution of keyword extraction techniques has seen a diverse range of methods spanning supervised and unsupervised paradigms. Supervised approaches, such as classification-based algorithms, leverage annotated datasets to train models capable of identifying keywords. Notable examples include KP-Miner (El-Beltagy and Rafea, 2009) and the supervised framework by (Papagiannopoulou and Tsoumakas, 2020). In contrast, unsupervised methods, which do not rely on labeled data, have predominantly employed graph-based techniques. Algorithms such as TextRank (Mihalcea and Tarau, 2004), SingleRank (Wan and Xiao, 2008), and MultipartiteRank (Boudin, 2018) utilize word co-occurrence graphs to rank and extract keywords. Additionally, TopicRank (Bougouin et al., 2013) and PositionRank (Florescu and Caragea, 2017) introduced refinements to graph-based methods by incorporating topical and positional information. Despite their effectiveness, these traditional methods often struggle with capturing nuanced and contextual information, limiting their applicability in more complex scenarios. In recent years, embedding-based techniques have significantly advanced keyword extraction by leveraging dense vector representations of words and phrases. EmbedRank (Bennani-Smires et al., 2018), for instance, employs Word2Vec (Mikolov, 2013) and Sent2Vec (Pagliardini et al., 2017) to generate embeddings for candidate phrases, which are then ranked based on cosine similarity with the document's representation. Building on these foundations, more recent methods like PatternRank and KeyBERT have integrated contextual embeddings derived from advanced language models such as SBERT and BERT (Schopf et al., 2022; Grootendorst, 2020). These approaches also incorporate syntactic patterns, such as Part-of-speech (PoS) tagging, to refine candidate phrase selection and improve contextual relevance. While these methods represent a substantial shift towards contextual keyword extraction, their reliance on predefined patterns and embeddings highlights the need for further advancements, particularly in harnessing the capabilities of LLMs. In this regard, (Boudin and Aizawa, 2024) proposed SILK, an unsupervised domain adaptation method leveraging citation contexts to synthesize training data, addressing the scarcity of annotated in-domain keyphrases. Concurrently, (Wu et al., 2024) introduced MetaKP, a paradigm for on-demand keyphrase generation guided by user intents, combining supervised fine-tuning and LLM-based prompting to handle dynamic goals. These works collectively advance

keyphrase generation using LLMs, demonstrating the field's shift toward flexible, resource-efficient solutions.

3 Dataset Construction

The multilingual dataset used for this study is constructed from the HAL database platform, an open archive dedicated to disseminating scientific research publications in French and English. Recent works, such as HALvest (Kulumba et al., 2024), demonstrate the underutilized potential of the HAL database for exploring and analyzing scientific publications. This dataset covers various scientific domains and its articles are accompanied by various information such as abstracts and author-provided keywords. We use abstracts, titles, and author-provided keywords, which will serve as a reference for evaluating the quality of the extraction methods. This dataset was compiled using a script that leveraged the HAL API. The collected data included approximately 12,000 articles. An initial sorting eliminated 1,300 duplicates, while about 6,000 other articles were excluded due to the absence of keywords or abstracts. After this filtering, the final corpus consists of 4,700 usable articles, representing about 30% of the initial data. An initial observation reveals a marked linguistic distribution with 85% of the articles in English and 15% in French. Regarding the English articles, the average number of keywords per article is 5.35, with an average keyword length of 2.14 words. In comparison, for French articles, the average number of keywords is slightly higher at 6.32, with an average length of 2.23 words.

The distribution of scientific domains also varies by language, as illustrated in Figure 2. Unsurprisingly, computer science remains the majority for both languages. Humanities rank second in French, while life sciences take this position in English. Humanities, well-represented in French, are less present in English. For the rest of the analysis, it is important to note that all titles, keywords, and abstracts were converted to lowercase to ensure consistent and reliable results.

4 Method

In this study, we approach keyword extraction through two distinct paradigms: *Generative Approaches* and *Embedding-Based Approaches*. For generative methods, we employ LLMs in a zero-shot learning framework, selected for its implementation simplicity and proven effectiveness in capturing baseline model performance. Formally, given an input document $D = \{w_1, ..., w_n\}$, the model generates candidate keywords K_G through conditional probability:

$$P(k|D) = \prod_{t=1}^{m} P(k_t|k_{<t}, D) \quad (1)$$

where $k \in K_G$ represents a generated keyword sequence of length m. The instruction prompt is a follows.

> **Instruction:** *As a keyword extraction master, your only mission here is to extract only the most relevant keywords that are present in the text. Put the list of keywords between brackets, comma-separated. DO NOT write something else than the keywords you're supposed to extract from the text. Skip the preamble and provide only the keywords. The text:{text}*

The embedding-based approach operates by measuring semantic similarity between document embeddings $\mathbf{e}_D$ and keyword embeddings $\mathbf{e}_k$ from a predefined vocabulary $\mathcal{V}$, using cosine similarity. Keywords $K_E = \{k \in \mathcal{V} | \text{sim}(D, k) \geq \tau\}$ are selected through thresholding at τ. Our implementation leverages KeyBERT, a BERT-based framework that identifies document-subphrase alignment through this similarity measure. The system employs two distinct keyword selection strategies governed by:

Maximal Marginal Relevance (MMR) Balances keyword relevance and diversity through a trade-off parameter $\lambda \in [0, 1]$:

$$k_i = \arg\max_{k \in \mathcal{V} \setminus K_E} \Big[\lambda \cdot \text{sim}(D, k) - (1 - \lambda) \cdot \max_{k_j \in K_E} \text{sim}(k, k_j) \Big] \quad (2)$$

Max Sum Distance (MSum) : To diversify the results, it takes the 2 x top-n most similar words/phrases to the document. Then, it takes all top-n combinations from the 2 x top-n words and extract the combinations that are the least similar to each other by cosine similarity.

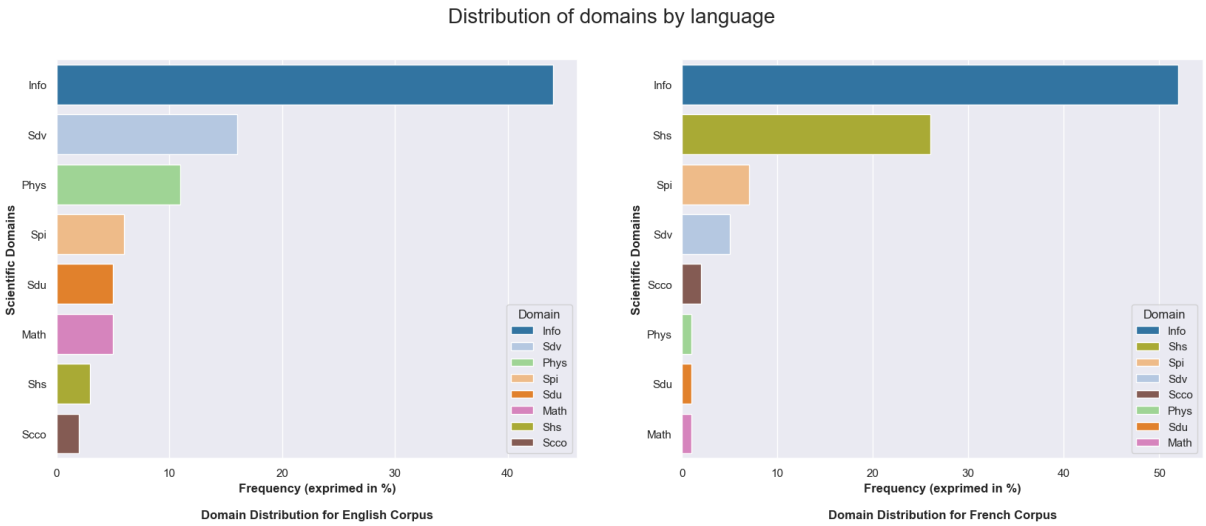


Figure 2: **Distribution of domains by language**

5 Experimental setup

In our study, we adopt an automatic evaluation framework to assess the performance of keyword extraction methods by comparing system-generated keywords against author-provided keywords from articles in the HAL open-access repository.

5.1 Methods

We evaluate three distinct categories of models. The first comprises multilingual LLMs that generate keywords in a generative manner, leveraging their pre-trained cross-lingual capabilities to produce contextually relevant terms. The second approach involves embedding-based models, where pre-trained embeddings encode textual content into dense vector representations, followed by clustering algorithms to identify salient keywords. The third category encompasses traditional statistical methods, which rely on frequency-based metrics, co-occurrence patterns, or graph centrality measures to extract candidate keywords.

Large Language Models The study leverages a diverse array of LLMs to ensure comprehensive evaluation across model architectures, scales, and accessibility frameworks. Open-weight models, chosen for their reproducibility and adaptability, include Meta's LLaMA 3.1 in both 70B and 8B parameter configurations, Mistral 7B, Mixtral 8x7B, and Google's Gemma 7B. These contrast with proprietary, closed-source models accessed via API, such as OpenAI's GPT-4o and GPT-3.5 Turbo, alongside Anthropic's Claude 3 Haiku and Claude Instant 1.2.

Embedding-based Models Our embedding-based approach employs KeyBERT, which utilizes pre-trained BERT embeddings to identify keywords by measuring semantic similarity between candidate terms and the input document. We evaluate two configurations: (1) a default setup relying solely on cosine similarity between document and keyword embeddings, and (2) an enhanced variant incorporating MMR for diversification and MSum to refine keyword selection by balancing relevance and novelty.

Traditional Models To establish robust baselines against contemporary neural approaches, we evaluate traditional unsupervised keyword extraction methods that rely on graph-based and statistical paradigms. This includes TextRank, a widely cited graph algorithm leveraging co-occurrence networks with PageRank-style scoring; PositionRank and SingleRank, which integrate term positional bias and heterogeneous graph structures, respectively; MultipartiteRank, optimized for topic-focused keyphrase extraction through multipartite graph representation; TopicRank, which hierarchically clusters candidate terms into topics before ranking; and YAKE, a lightweight statistical method combining term frequency, casing, and positional features.

5.2 Metrics

The comparison is performed within two approaches: **(1) Exact Matching**, where extracted keywords are evaluated based on their relevance

Model	Abstract + Title			Abstract		
	Precision	Recall	F1	Precision	Recall	F1
LLM-based Approach						
LLaMA 3.1 70b	0.132	0.245	**0.163**	0.120	0.224	**0.148**
Claude 3 Haiku	0.130	0.218	0.154	0.120	0.204	0.143
LLaMA 3.1 8b	0.147	0.181	0.151	0.136	0.172	0.142
GPT 4o	0.075	0.222	0.108	0.071	0.206	0.101
Claude Instant 1.2	0.073	0.183	0.097	0.066	0.171	0.088
GPT 3.5 Turbo	0.089	0.094	0.087	0.086	0.089	0.083
Mixtral 8x7b	0.057	0.188	0.083	0.047	0.176	0.070
Mistral 7b	0.050	0.199	0.077	0.048	0.156	0.069
Gemma 7b	0.051	0.079	0.059	0.052	0.081	0.060
Embedding-based Approach						
KeyBERT Default	0.058	0.081	0.067	0.056	0.078	0.065
KeyBERT with MMR and MSum	0.052	0.073	0.061	0.050	0.070	0.058
Traditional Approach						
PositionRank	0.062	0.115	0.080	0.056	0.103	0.072
MultipartiteRank	0.062	0.113	0.079	0.056	0.103	0.072
TopicRank	0.059	0.108	0.076	0.053	0.096	0.068
SingleRank	0.053	0.098	0.068	0.052	0.096	0.067
YAKE	0.053	0.098	0.068	0.045	0.083	0.058
TextRank	0.039	0.072	0.050	0.036	0.066	0.046

Table 1: **Evaluation Result with Exact Matching**

and precision compared to the keywords provided by the authors in their articles. The evaluation criteria include precision, recall, and the F1 measure. **(2) Fuzzy Matching**, which is a less strict method of term comparison without tolerance for variations such as plural forms, hyphen usage, or potential typographical errors.

Exact Matching In this approach, only identical terms were considered matches, to ensure a precise and consistent evaluation of the results. For each article, the most relevant keywords are extracted from the abstracts using all evaluated methods. We use the F1-Score, a commonly employed metric for evaluating the performance of keyword extraction models. The F1-Score is the harmonic mean between precision, which is the ratio of correctly extracted keywords to the total number of extracted keywords, and recall, which measures the proportion of relevant extracted keywords to the total number of relevant keywords in the text. In the context of keyword extraction, a high F1-Score indicates that the model successfully extracts a significant proportion of relevant keywords (high recall) while limiting the extraction of irrelevant keywords (high precision).

Fuzzy Matching This approach allows comparing generated keywords with reference keywords by considering formal variations. Several metrics can assign a "proximity score" between two strings, such as Levenshtein, Jaro-Winkler, and various embedding models (Alqahtani et al., 2021). In this study, we adopt the Levenshtein distance, also known as edit distance. It quantifies the minimum number of operations required to transform one string into another, with possible operations being insertion, deletion, or substitution of characters. The results are presented in graphical form to illustrate the evolution of the F1-Score as the flexibility of the Levenshtein distance increases (from 0 to 4).

6 Results

The evaluation results, as detailed in Table 1, compare model performance across precision, recall, and F1-score under two input settings: (1) Abstract With Title and (2) Abstract Only, ranked by decreas-

Model	Abstract + Title				Abstract			
	$d \leq 1$	$d \leq 2$	$d \leq 3$	$d \leq 4$	$d \leq 1$	$d \leq 2$	$d \leq 3$	$d \leq 4$
LLM-based Approach								
LLaMA 3.1 70b	**0.19**	**0.197**	**0.21**	**0.228**	**0.174**	**0.18**	**0.193**	**0.212**
Claude 3 Haiku	0.179	0.185	0.195	0.21	0.168	0.173	0.183	0.198
LLaMA 3.1 8b	0.175	0.183	0.198	0.223	0.165	0.172	0.187	0.21
GPT 4o	0.127	0.132	0.141	0.155	0.12	0.124	0.134	0.148
Claude Instant 1.2	0.116	0.13	0.147	0.176	0.105	0.118	0.135	0.163
GPT 3.5 Turbo	0.101	0.106	0.118	0.137	0.096	0.102	0.114	0.13
Mixtral 8x7b	0.1	0.107	0.118	0.133	0.084	0.095	0.107	0.123
Mistral 7b	0.092	0.097	0.107	0.123	0.085	0.09	0.099	0.116
Gemma 7b	0.069	0.072	0.076	0.084	0.071	0.073	0.078	0.086
Embedding-based Approach								
KeyBERT Default	0.084	0.095	0.12	0.158	0.081	0.092	0.116	0.154
KeyBERT with MMR and MSum	0.072	0.08	0.101	0.137	0.07	0.078	0.098	0.135
Traditional Approach								
PositionRank	0.097	0.101	0.108	0.123	0.087	0.091	0.099	0.114
MultipartiteRank	0.095	0.099	0.113	0.139	0.087	0.091	0.105	0.13
TopicRank	0.089	0.094	0.108	0.135	0.08	0.084	0.099	0.125
SingleRank	0.083	0.087	0.092	0.102	0.072	0.075	0.081	0.091
YAKE	0.081	0.085	0.094	0.113	0.079	0.082	0.091	0.11
TextRank	0.062	0.065	0.068	0.075	0.058	0.06	0.064	0.071

Table 2: **Evaluation Result with Fuzzy Matching (F1 Scores)**

ing effectiveness. Traditional graph-based methods exhibit stark disparities, with performance gaps exceeding 60% between the weakest (TextRank) and strongest models (PositionRank and MultipartiteRank). In contrast, KeyBERT demonstrates near-equivalent performance across both input variants, suggesting robustness to textual context. Notably, the inclusion of titles yields minimal impact on traditional and KeyBERT-based methods. However, LLMs display significant variability, with performance ranging from modest to triple-digit improvements when titles are included, boosting metrics by approximately 10%. The top-performing LLMs—LLaMA 3.1 70B, Claude 3 Haiku, and LLaMA 3.1 8B—highlight the role of scale and architecture in keyword extraction, while Gemma 7B's subpar performance underscores the criticality of prompt compliance, as deviations in output formatting led to severe penalties under exact-match evaluation.

The experimental findings, illustrated in Table 2, underscore the utility of Levenshtein distance in accommodating linguistic variations, which enhances precision at the cost of computational efficiency. While traditional models exhibit moderate performance gains when titles are included, KeyBERT demonstrates superior robustness in keyword extraction by leveraging contextual embeddings, particularly in texts with heterogeneous term distributions. This approach mitigates reliance on surface-level patterns, offering nuanced semantic alignment. LLMs, capitalizing on their deep contextual awareness and capacity to process structurally diverse texts, consistently outperform alternative methods, especially in complex extraction tasks. Generative architectures further benefit from the flexibility of Levenshtein-based evaluation, though title inclusion yields diminishing returns beyond a performance threshold. These results highlight a critical trade-off: while Levenshtein distance and contextual embeddings improve precision and adaptability, they introduce computational overhead. The interplay between model architecture, input context (e.g., title inclusion), and evaluation metrics emerges as a pivotal factor in optimizing keyword extraction systems, with LLMs setting

a high benchmark for accuracy despite scalability challenges.

7 LLMs and Cost per Token

The computational and environmental costs of LLMs present critical barriers to accessibility and sustainability, particularly for institutions with limited resources. As evidenced by our analysis, models achieving comparable F1 scores can vary by 10–100x in operational costs per token, underscoring the need to integrate economic and ecological considerations into evaluation frameworks. To address this gap, we propose the Token Efficiency Score (TES), a novel metric balancing performance (F1) and cost ($/million tokens) through a weighted harmonic mean that prioritizes affordability without sacrificing accuracy. The formula,

$$\text{TES} = \frac{(1+\alpha) \times F_1 \times \text{Cost}}{\alpha \times \text{Cost} + F_1} \quad (\alpha = 10), \quad (3)$$

applies a strong penalty to cost, reflecting its outsized impact in mass data processing scenarios. While LLMs excel in task performance, their resource intensity highlights a critical trade-off: high-parameter models like GPT-4 achieve marginal gains at prohibitive expense, whereas smaller models (e.g., LLaMA-7B) offer viable efficiency-performance equilibria. TES not only democratizes model selection for resource-constrained environments but also incentivizes energy-conscious development, aligning AI progress with sustainability goals. This metric redefines evaluation paradigms, urging the community to prioritize computational equity alongside technical prowess—a crucial step toward ethical, scalable NLP solutions.

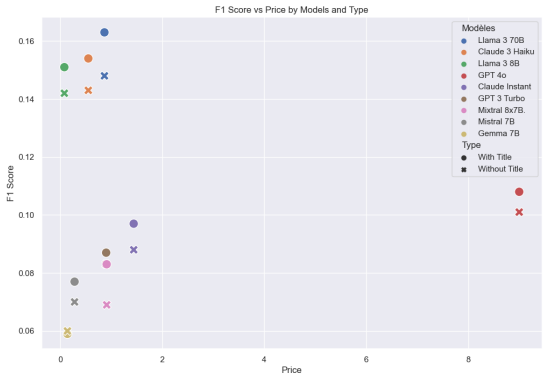
The calculation shows that the most performant models are also among the least costly, notably Llama-3 70B, Llama-3 8B, and Claude 3 Haiku. As shown in Figure 3, we rank the generative models by their TES score from most efficient to least efficient. As expected, the top three models are Llama 3 70B, Claude 3 Haiku, and Llama 3 8B, with Gemma 7B by Google in the last position. The TES allows for clear identification of the most performant models while considering the cost factor, which is crucial in large-scale scenarios.

8 Limitations

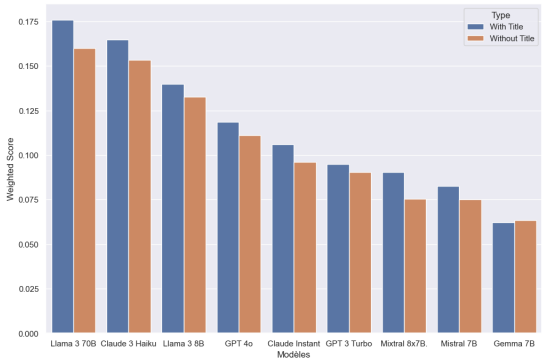
While LLMs have revolutionized keyword extraction through their contextual depth and adaptability, their deployment in scientific settings reveals critical limitations. First, their reliance on generic pretraining corpora restricts domain-specific precision, necessitating costly fine-tuning on annotated technical datasets to capture discipline-specific terminology. Second, their inherent opacity as "black-box" systems complicates interpretability, hindering traceability in scenarios requiring explainable keyword selection processes. Third, LLMs exhibit stochastic instability, with outputs fluctuating based on prompt phrasing—a challenge demanding iterative prompt engineering and repeated evaluations to stabilize F1-score performance. This instability is compounded by cost-efficiency trade-offs: verbose, conversational prompts may marginally improve keyword structure but inflate computational expenses without guaranteed gains in relevance. Finally, evaluation frameworks face intrinsic biases, exemplified by the HAL corpus, where absent keyword mentions in abstracts/titles disadvantage extractive models. These limitations underscore the need for domain-adapted training paradigms, standardized prompt templates, and evaluation corpora that align author-provided keywords with textual content—critical steps toward bridging the gap between LLM capabilities and scientific keyword extraction requirements.

9 Conclusion and Future Work

The experimental findings underscore the transformative potential of generative LLMs in keyword extraction, surpassing traditional methods in precision and semantic relevance, even in zero-shot settings. By capturing nuanced contextual relationships, LLMs produce keywords that better reflect scientific content, while our proposed Token Efficiency Score (TES) highlights cost-effective models—such as Claude 3 Haiku and LLaMA variants—that balance performance and affordability. Notably, integrating titles enhances F1-scores without significantly increasing computational overhead, emphasizing the value of metadata in extraction tasks. Future work should prioritize prompt engineering to stabilize outputs—for instance, by specifying keyword length or structuring prompts as simulated dialogues to reduce format variability, particularly for models like Gemma. Fine-tuning LLMs on domain-specific corpora could further bridge gaps between generative and extractive methods, while expanding processing to full-text articles (Teufel and Moens, 2002) promises



(a) F1 Score Performance relative to Price.



(b) Weighted Score

Figure 3: Cost and Weighted Score

richer keyword extraction by leveraging broader contextual signals. Complementing F1-score with metrics like NPMI and BM25 could better evaluate semantic coherence, and integrating thematic modeling (e.g., BERTopic) may organize keywords into structured taxonomies, enhancing interpretability. These directions not only refine extraction accuracy but also address scalability and domain adaptation challenges, laying the groundwork for LLMs to serve as versatile, sustainable tools for scholarly knowledge organization—a critical advancement as NLP increasingly intersects with scientific publishing and meta-research. This roadmap calls for interdisciplinary collaboration to align technical innovation with real-world usability and environmental responsibility.

Acknowledgment

We gratefully acknowledge the Sorbonne Center for Artificial Intelligence (SCAI) for partially funding this research.

References

Awatif Alqahtani, Hosam Alhakami, Tahani Alsubait, and Abdullah Baz. 2021. A survey of text matching techniques. *Engineering, Technology & Applied Science Research*, 11(1):6656–6661.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470*.

Santosh Kumar Bharti and Korra Sathya Babu. 2017. Automatic keyword extraction for text summarization: A survey. *arXiv preprint arXiv:1704.03242*.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. *arXiv preprint arXiv:1803.08721*.

Florian Boudin and Akiko Aizawa. 2024. Unsupervised domain adaptation for keyphrase generation using citation contexts. *arXiv preprint arXiv:2409.13266*.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551.

David B Bracewell, Fuji Ren, and Shingo Kuriowa. 2005. Multilingual single document keyword extraction for information retrieval. In *2005 international conference on natural language processing and knowledge engineering*, pages 517–522. IEEE.

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

Samhaa R El-Beltagy and Ahmed Rafea. 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information systems*, 34(1):132–144.

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

Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1105–1115.

Maarten Grootendorst. 2020. Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics. *Zenodo, Version v0*, 9(10.5281).

Fahrettin Horasan. 2021. Keyword extraction for search engine optimization using latent semantic analysis. *Politeknik Dergisi*, 24(2):473–479.

Daehoon Kim, Daeyong Kim, Eenjun Hwang, and Seungmin Rho. 2015. Twittertrends: a spatio-temporal trend detection and related keywords recommendation scheme. *Multimedia Systems*, 21:73–86.

Francis Kulumba, Wissam Antoun, Guillaume Vimont, and Laurent Romary. 2024. Harvesting textual and structured data from the hal publication repository. *arXiv preprint arXiv:2407.20595*.

Reza Yousefi Maragheh, Chenhao Fang, Charan Chand Irugu, Parth Parikh, Jason Cho, Jianpeng Xu, Saranyan Sukumar, Malay Patel, Evren Korpeoglu, Sushant Kumar, et al. 2023. Llm-take: Theme-aware keyword extraction using large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4318–4324. IEEE.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2020. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1339.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Gerard Salton and Chris Buckley. 1990. A note on term weighting and text matching. Technical report, Cornell University.

Tim Schopf, Simon Klimek, and Florian Matthes. 2022. Patternrank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. *arXiv preprint arXiv:2210.05245*.

Mingyang Song, Yi Feng, and Liping Jing. 2023a. A survey on recent advances in keyphrase extraction from pre-trained language models. *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2153–2164.

Mingyang Song, Xuelian Geng, Songfang Yao, Shilong Lu, Yi Feng, and Liping Jing. 2023b. Large language models as zero-shot keyphrase extractors: A preliminary empirical study. *arXiv preprint arXiv:2312.15156*.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.

Di Wu, Xiaoxian Shen, and Kai-Wei Chang. 2024. Metakp: On-demand keyphrase generation. *arXiv preprint arXiv:2407.00191*.