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

Nacef Ben Mansour[1] Hamed Rahimi[2] Motasem Alrahabi[3]

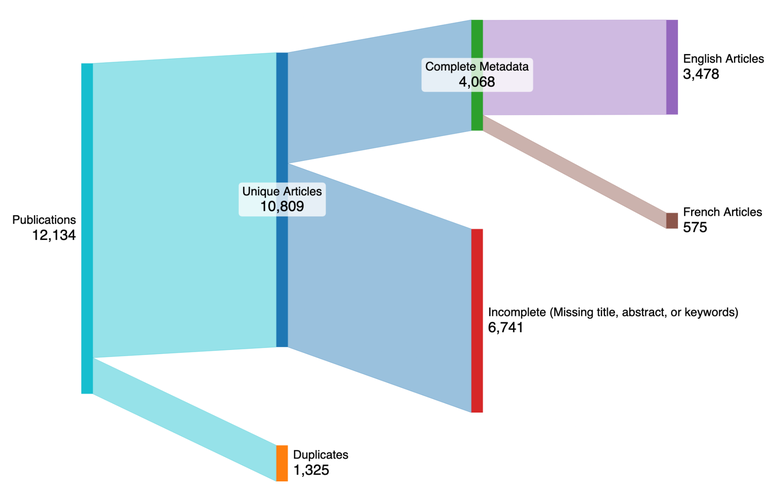
[1]Sorbonne Université, Paris, France [2]ISIR, Sorbonne Université, Paris, France [3]ObTIC, Sorbonne Université, Paris, France

Summary

We compare traditional keyword extraction methods (like TextRank and YAKE) with LLM across scientific articles. We evaluate using F1 score for both exact matching and fuzzy matching, where Levenshtein distance thresholds capture variation in keyword forms. LLMs consistently outperform traditional methods in both precision and relevance, highlighting their potential in enhancing scientific information retrieval. More surprisingly, we find that smaller, cheaper LLMs often outperform larger, more expensive ones—achieving better accuracy at a fraction of the cost.

Dataset & Preprocessing

We constructed a multilingual dataset from the HAL open archive using the HAL API, retrieving approximately 12,000 articles.
After removing 1,300 duplicates and entries lacking abstracts or keywords, we retained 4,700 articles (∼30% of the original set).



Data filtering and preprocessing pipeline

All text (titles, abstracts, keywords) was lowercased for consistency.

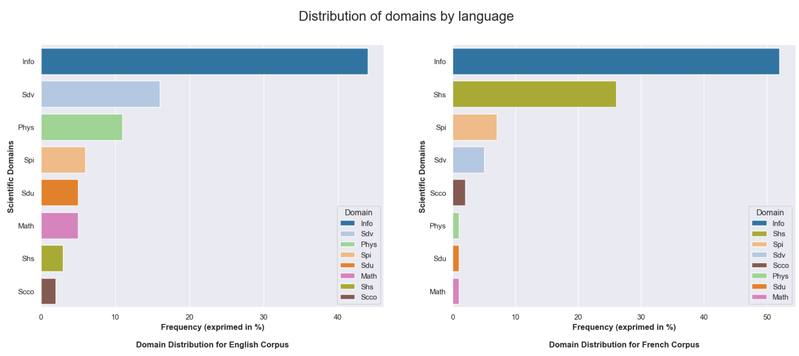
Data Analysis

Language Distribution and Keyword Statistics :

- 85% English, 15% French
- **English:** 5.35 keywords per article & 2.14 words per keyword
- **French**: 6.32 keywords per article & 2.23 words per keyword

Scientific Domain Trends

- Computer Science dominates both languages
- Humanities are more prevalent in French
- Life Sciences rank second in English



Scientific domains by language

Results

Model	Abstract + Title			Abstract		
	P	R	F1	P	R	F1
LLM-based						
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						
KeyBERT Default	0.058	0.081	0.067	0.056	0.078	0.065
KeyBERT+MMR/MSum	0.052	0.073	0.061	0.050	0.070	0.058
Traditional						
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: Exact Matching Scores

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								
LLaMA 70b	**0.190**	**0.197**	**0.210**	**0.228**	**0.174**	**0.180**	**0.193**	**0.212**
Claude 3	0.179	0.185	0.195	0.210	0.168	0.173	0.183	0.198
LLaMA 8b	0.175	0.183	0.198	0.223	0.165	0.172	0.187	0.210
GPT-4o	0.127	0.132	0.141	0.155	0.120	0.124	0.134	0.148
Claude Inst	0.116	0.130	0.147	0.176	0.105	0.118	0.135	0.163
GPT-3.5	0.101	0.106	0.118	0.137	0.096	0.102	0.114	0.130
Mixtral	0.100	0.107	0.118	0.133	0.084	0.095	0.107	0.123
Mistral	0.092	0.097	0.107	0.123	0.085	0.090	0.099	0.116
Gemma	0.069	0.072	0.076	0.084	0.071	0.073	0.078	0.086
Embedding-based								
KeyBERT	0.084	0.095	0.120	0.158	0.081	0.092	0.116	0.154
KeyBERT+MMR	0.072	0.080	0.101	0.137	0.070	0.078	0.098	0.135
Traditional								
PositionRank	0.097	0.101	0.108	0.123	0.087	0.091	0.099	0.114
Multipartite	0.095	0.099	0.113	0.139	0.087	0.091	0.105	0.130
TopicRank	0.089	0.094	0.108	0.135	0.080	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	0.110
TextRank	0.062	0.065	0.068	0.075	0.058	0.060	0.064	0.071

Table: Fuzzy Matching (F1 across distances)

Keyword Extraction Methods

We evaluate three families of keyword extraction approaches:

Large Language Models (LLMs):

- **Closed Source:** GPT-4o, GPT-3.5, Claude 3 Haiku
- **Open Source:** LLaMA 3.1 (70B, 8B), Mixtral, Mistral, Gemma

Embedding-based Models:

- Based on KeyBERT using BERT embeddings

Traditional Models:

- TextRank, PositionRank, SingleRank, MultipartiteRank, TopicRank, YAKE

Prompt used

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}*

Evaluation Metrics

Two Evaluation Settings:

- **Exact Matching:** Strict term-level match
- **Fuzzy Matching:** Several Levenshtein distance thresholds

Main Metric: F1-Score

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

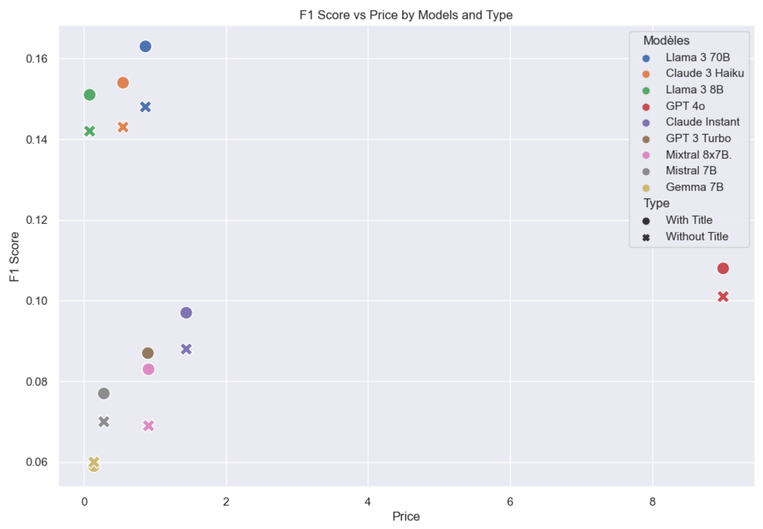
Token Efficiency Score (TES):

$$\text{TES} = \frac{(1+\alpha) \times \text{F1} \times \text{Cost}}{\alpha \times \text{Cost} + \text{F1}} \quad (\alpha = 10) \quad (2)$$

Weighted harmonic mean of F1 and inference cost per millions of token.

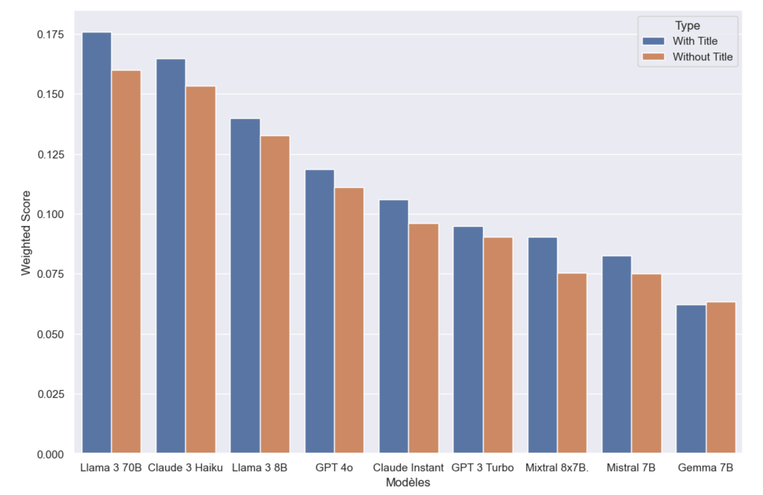
LLMs and Cost per Token

Our analysis shows that models with similar F1 scores can differ in cost by **10–100×**.



F1 Score Performance relative to Price

TES shows that models such as **LLaMA 3 70B**, **Claude 3 Haiku**, and **LLaMA 3 8B** outperform larger models like **GPT-4** *while operating at just a fraction of the cost.*



TES per LLM

Supplementary Materials

Scan the QR code to access the full paper and code.