Methods and Evaluation in Unsupervised Keyphrase Prediction: A Survey

Yuchen Han and Huiqian Wu and Yuqing Sun(✉)

Shandong University, Jinan 250000, China
hanyc@mail.sdu.edu.cn, w50305w@163.com, sun_yuqing@sdu.edu.cn

Abstract. Keyphrases are concise semantic units that capture the meanings of a document, providing interpretable abstractions that reduce textual complexity and support downstream applications such as information retrieval, text summarization, and document classification. While supervised keyphrase prediction methods rely on costly annotated data, unsupervised methods have emerged as effective alternatives. This survey provides a comprehensive overview of unsupervised keyphrase prediction methods and systematically reviews the evaluation metrics. We analyze the linguistic properties of keyphrases to inform the design of evaluation metrics and guide the improvement of prediction methods. In addition, we categorize existing unsupervised approaches and discuss their respective strengths and limitations. Furthermore, we review existing evaluation metrics and analyze their shortcomings. To address these gaps, we introduce CAME, a novel reference-free metric and validate its effectiveness through experiments. Finally, we highlight challenges and promising directions, particularly in the context of LLMs, to guide future research in unsupervised keyphrase prediction.

Keywords: Unsupervised Method · Keyphrase Prediction · Context-Aware Multi-dimensional Evaluation.

1 Introduction

Keyphrase Prediction (KP) aims to automatically identify a set of phrases that concisely represent the core semantic content of a document, thereby facilitating efficient information organization, retrieval, and knowledge discovery. Keyphrases can be broadly categorized into two types: present keyphrases, which explicitly occur in the source text [44], and absent keyphrases, which are semantically related to the document but exhibit no or only partial lexical overlap with it [40] (Fig. 1). Present keyphrases reflect the explicit content of the document, whereas absent keyphrases reveal implicit or inferred concepts that are not directly mentioned in the text. Both types of keyphrases are essential, as high-quality keyphrases not only facilitate content summarization but also improve document–query matching in information retrieval tasks [10]. For instance, in a scholarly article discussing the application of machine learning in medical image analysis, technical terms such as “convolutional neural network” and “image segmentatio” can serve as present

TEXT: "The nitrogen-vacancy (NV) center is a point defect in diamond with unique properties for use in ultra-sensitive, high-resolution magnetometry. One of the most interesting and challenging applications is nanoscale magnetic resonance imaging (nano-MRI). While many review papers have covered other NV centers in diamond applications, there is no survey targeting the specific development of nano-MRI devices based on NV centers in diamond. Several different nano-MRI methods based on NV centers have been proposed with the goal of improving the spatial and temporal resolution, but without any coordinated effort. After summarizing the main NV magnetic imaging methods, this review presents a survey of the latest advances in NV center nano-MRI. "

PRESENT Keyphrases: [" nitrogen-vacancy center ", " nanoscale magnetic resonance imaging (nano-MRI) "]

ABSENT Keyphrases: ["nanodiamonds", "optically detected magnetic resonance"]

Fig. 1: An example from Kp-Biomed [24] dataset. The present keyphrases are highlighted, while the absent keyphrase 'optically detected magnetic resonance' does not appear in the source document.

keyphrases, highlighting the paper's methodological contributions. Meanwhile, domain-relevant phrases such as "AI-assisted diagnosis" and "intelligent medical image analysis", though not explicitly mentioned in the text, are categorized as as absent keyphrases to enhance content summarization and improve the article's visibility in search results.

Supervised neural network models can capture the implicit semantics of documents, which enhances keyphrase prediction. Traditional keyphrase extraction methods treat a document as a bag of words and extract present keyphrases based on expert-defined rules [12]. To better capture the structural relationships among words, graph-based approaches model word interactions, allowing for more accurate estimation of phrase relevance within the document.

However, those supervised models rely on high-cost labeled data and perform suboptimally in out-of-domain documents [18]. Therefore, unsupervised keyphrase prediction represents an attractive alternative, providing enhanced flexibility and improved domain generalization.

Unsupervised keyphrase prediction is particularly valuable in domains where annotated data is scarce or costly to obtain. Early unsupervised methods primarily relied on syntactic and statistical features. Despite their simplicity and computational efficiency, these approaches depend on surface-level cues, such as term frequency and part-of-speech patterns, and often struggle to capture deeper semantic relationships.

To address these limitations, graph-based methods effectively capture the structural and relational information within a document, leading to improved keyphrase selection. However, graph-based methods still rely on heuristic rules for graph construction and may struggle to fully capture deep semantic meaning or generalize across domains.

Recent advances in unsupervised keyphrase prediction have been driven by pretrained language models (PLMs), particularly large language models (LLMs). With strong text understanding and generation capabilities, LLMs have demonstrated impressive zero-shot performance across NLP tasks. Techniques such as prompt-based learning further enhance their effectiveness, leading to significant improvements in unsupervised keyphrase prediction. However, LLMs face notable challenges in domain-specific contexts: outdated or incomplete

knowledge can hinder the generation of high-quality absent keyphrases, sensitivity to prompt design affects performance, and hallucination may produce redundant, verbose, or incomplete keyphrase sets. Furthermore, the lack of reference-free evaluation metrics hampers the iterative refinement of LLM-generated keyphrases.

Motivated by these challenges, this survey offers a comprehensive analysis of unsupervised keyphrase prediction (Fig. 2). The key contributions of this work are summarized as follows:

- We introduce a novel linguistic perspective on keyphrases, systematically analyzing their intrinsic properties. This perspective not only informs the design of more comprehensive evaluation metrics but also provides targeted guidance for the development of prediction methods.
- We provide a systematic categorization of existing unsupervised keyphrase prediction methods and conduct a comparative analysis of their strengths and limitations.
- We analyze existing keyphrase evaluation metrics from the perspective of the intrinsic linguistic properties of keyphrases, summarize their limitations, and propose CAME, a novel reference-free metric designed to enhance the robustness of keyphrase evaluation.
- We summarize the key challenges in unsupervised keyphrase prediction and outline promising directions for future research.

2 Overview

The remainder of this paper is organized as follows. Section 2 provides an overview of the keyphrase prediction task and highlights the properties of keyphrases. Sections 3 and 4 review unsupervised methods for keyphrase extraction and generation, respectively. Section 5 examines existing evaluation metrics and Section 6 introduces the proposed reference-free metric, **CAME**, and validates its effectiveness. Section 7 discusses promising directions for future research, and Section 8 concludes the survey.

2.1 Keyphrase Property Analysis from a Linguistic Perspective

To inform the design and evaluation of keyphrase prediction, we examine the syntactic and semantic properties of the keyphrases produced by these models. As illustrated in Fig. 2, the generated keyphrases should satisfy four key properties: Conciseness, Coverage, Distinctiveness, and Diversity.

Syntactic Properties Syntactic properties describe the grammatical structure of keyphrases. As shown in Fig. 1, present keyphrases are phrases that appear verbatim as contiguous spans in the source document, whereas absent keyphrases are not explicitly mentioned and must be inferred from the document content (e.g., by combining non-consecutive textual elements or summarizing higher-level topics). According to the analysis in [40], present keyphrases account for 55.69%

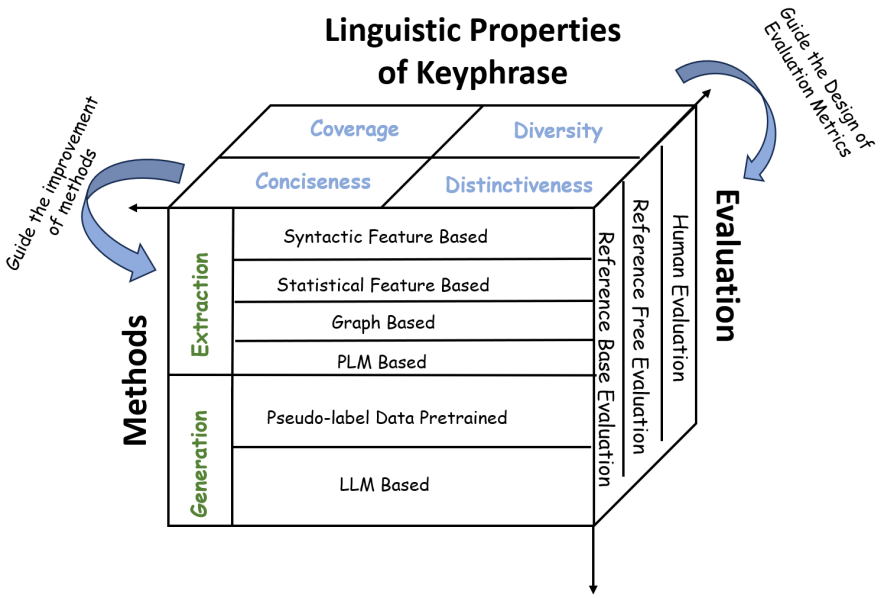


Fig. 2: Motivation: Role of Linguistic Perspective in Keyphrase Research

in the Inspec dataset [25], 44.74% in Krapivin2009 [32], 67.75% in NUS [45], and 42.01% in SemEval [30], with the remainder being absent. Later work [9] refined this binary distinction by introducing a four-class taxonomy of absent keyphrases, which subsequently informed the construction of the biomedical dataset Kp-Biomed [24].

Secondly, keyphrases are not bound by fixed length or strict boundaries, but are generally formed by zero or more adjectives followed by one or more nouns. They should retain the original lexical forms from the source text, avoiding redundancy or unnecessary extension. Thus, the first property of a keyphrase can be defined as conciseness.

Conciseness *refers to the quality of a keyphrase being as succinct as possible while faithfully reflecting the source content. It should be a noun phrase that does not include redundant words or omit essential modifiers [19].* For instance, extending 'Chromophore-assisted light inactivation' to 'Chromophore-assisted light inactivation method' introduces redundancy, as 'method' does not enhance thematic relevance. Conversely, truncating it to 'light inactivation' removes the crucial modifier 'Chromophore-assisted', thereby distorting the intended meaning.

Semantic Properties Semantic properties concern the meaning that keyphrases are expected to convey. Prior research [19, 69] suggests that an effective keyphrase set should both capture the essential content of a document and maintain distinctiveness across documents. This differs from 'high-quality phrases' in phrase

mining, which primarily emphasize domain-level popularity rather than document-specific relevance.

Coverage *refers to how well a set of keyphrases collectively captures the main topics and essential content of the document [64].* From the perspective of the relationship between keyphrases and the source document, coverage ensures that essential content is retained when compressing a long text into keyphrases. While some information loss is unavoidable, a good keyphrase set minimizes this loss and encourages the inclusion of absent keyphrases, which can provide higher-level summaries.

Diversity *refers to the extent to which the keyphrase set avoids semantic redundancy and ensures information complementarity among keyphrases [69].* From the perspective of relationships within a keyphrase set, diversity requires minimizing duplication while maximizing unique contributions. A high-quality set balances present keyphrases, which capture localized details, and absent keyphrases, which summarize broader themes.

Distinctiveness *refers to a keyphrase's ability to distinguish the current document from others, either within the same domain or across domains[?].* From the perspective of the relationship between keyphrases and the domain, distinctiveness is weakened when phrases are overly generic. For instance, words like "protei" and "cell" are ubiquitous in the biomedical field, whereas more specific expressions such as "cell signaling" are less frequent and provide better discrimination. As keyphrase prediction increasingly targets domain-specific texts, distinctiveness becomes crucial for supporting specialized content analysis.

As shown in Fig. 2, the linguistic perspective of keyphrases not only informs the design of evaluation metrics but also provides targeted guidance for method development.

2.2 Taxonomies of Unsupervised Keyphrase Prediction Methods

Keyphrase prediction can be categorized into two types according to whether the keyphrases appear in the source text: (1) Keyphrase Extraction (KPE), which targets only present keyphrases, and (2) Keyphrase Generation (KPG), which produces both present and absent keyphrases. As shown in Fig. 2, each of these tasks can be further divided into finer categories, reflecting the methodological diversity of unsupervised approaches.

For Keyphrase Extraction, we further categorize methods into four classes: (1) Syntactic feature based methods, (2)Statistical feature based methods, (2) Graph based Unsupervised Keyphrase Extraction, and (3) Pretrained Language Model (PLM) based Unsupervised Keyphrase Extraction. This categorization reflects the historical evolution and methodological diversity of the field: early methods relied on surface-level syntactic or statistical features, graph-based methods introduced structural relationships among words to improve keyphrase selection, and PLM-based approaches leverage the semantic understanding of large pretrained models to capture both explicit and implicit keyphrases.

For Keyphrase Generation, we categorize unsupervised methods into two main classes: (1) Pseudo Data Based Unsupervised Keyphrase Generation, which

relies on automatically generated pseudo-labeled data to train models without human annotation, and (2) Large Language Model (LLM) Based Unsupervised Keyphrase Generation, which leverages the strong text understanding and generation capabilities of pretrained LLMs to produce both present and absent keyphrases. Organizing these methods in this way helps to clarify their underlying assumptions, strengths, and limitations, providing a structured perspective for both comparison and further research.

3 Unsupervised Keyphrase Extraction Methods

Fig. 3 provides a comprehensive overview of unsupervised keyphrase extraction and generation methods.

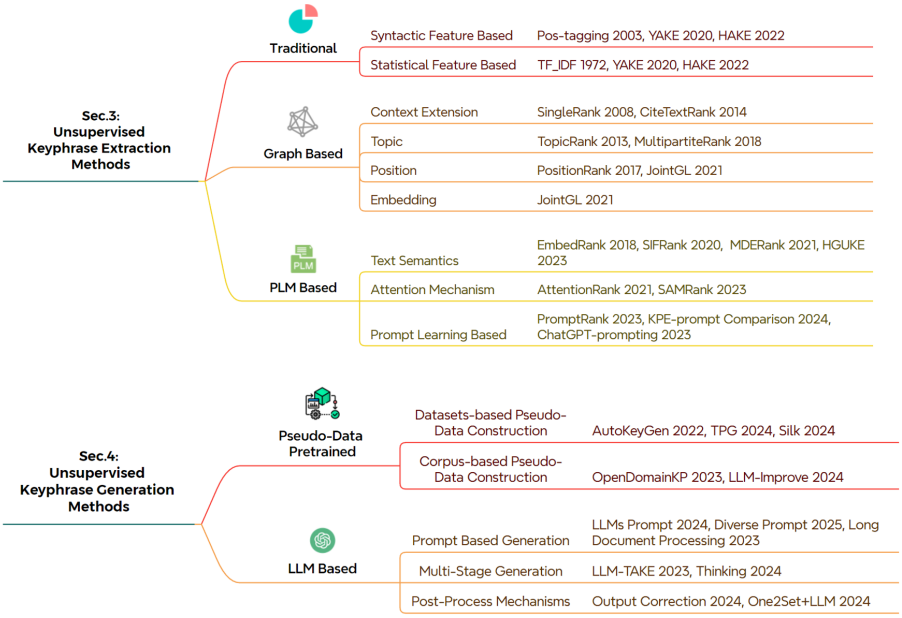


Fig. 3: Overview of Unsupervised Keyphrase Extraction and Generation Methods.

3.1 Syntactic Feature Based Unsupervised Keyphrase Extraction

Keyphrases are typically realized as noun phrases, consisting of one or more nouns possibly preceded by one or more adjectives. They can be captured using the part-of-speech pattern <NN.|JJ> <NN.*>, where "NN" denotes nouns and "JJ" denotes adjectives. PoS-tagging [25] identifies potential keyphrases by defining frequent part-of-speech patterns (e.g., adjective + noun, noun + noun) and extracting text spans that conform to these patterns. The process often involves

tokenizing the text, tagging parts of speech using StanfordCoreNLP tools[1], and selecting phrases that match the pattern using NLTK[2]. In addition, PoS tags can be incorporated as salient features in machine learning classifiers, thereby improving the distinction between keyphrases and non-keyphrases.

YAKE [12] relies on a combination of statistical features computed directly from the input document, such as term position, frequency, casing, and word relatedness. HAKE [41] leverages syntactic information such as dependency parsing and part-of-speech tagging to automatically identify high-quality keyphrases from a single document. By incorporating grammatical relations, it is able to detect candidate phrases beyond simple n-gram patterns, ensuring that the extracted keyphrases are syntactically valid.

Syntactic feature based methods are generally efficient, interpretable, and broadly applicable. Since they rely primarily on part-of-speech patterns and simple statistical cues, they can be implemented with relatively low computational cost and yield transparent results that are easy to analyze.

However, these approaches also have notable drawbacks. They often ignore deeper contextual, structural, and semantic information, which limits their ability to capture the nuanced meaning of terms. Moreover, they provide limited support for multi-word expressions and synonymous phrases: key concepts expressed in different lexical forms may be missed, and phrases with subtle semantic differences may not be distinguished.

3.2 Statistical Feature Based Unsupervised Keyphrase Extraction

Statistical feature–based methods exploit surface-level distributional patterns of words and phrases to estimate their importance. A classical measure is TF-IDF [60], where words occurring frequently in a document but infrequently across the corpus are considered more representative of the document's semantics. Owing to its simplicity and effectiveness, TF-IDF has been widely adopted in unsupervised keyphrase extraction, either directly or as part of feature engineering pipelines. For example, the AutoKeyGen model [53] utilizes TF-IDF to generate pseudo-labels, guiding downstream training without requiring human annotations.

Beyond term frequency statistics, co-occurrence features have been shown to be valuable for capturing the contextual associations between words. Words that frequently appear together within a sliding window or syntactic neighborhood are more likely to form meaningful phrases. Measures such as Pointwise Mutual Information (PMI)[15], and word association strength [6] have been extensively applied to rank candidate keyphrases. These approaches allow statistical methods to move beyond unigrams and incorporate phrase-level salience.

Recent work often combines syntactic and statistical cues to overcome the limitations of relying on a single feature source. For instance, YAKE [12] incorporates multiple document-level statistical signals, including word frequency, position, casing, and word relatedness, to robustly estimate term importance. HAKE [41],

[1] https://stanfordnlp.github.io/CoreNLP/
[2] https://github.com/nltk

in contrast, integrates statistical evidence with syntactic information such as dependency parsing, enabling the identification of longer and syntactically valid keyphrases. These hybrid methods are efficient, domain-agnostic, and capable of capturing phrase salience from multiple perspectives, making them a strong baseline in unsupervised keyphrase prediction.

Despite their advantages, statistical methods generally lack semantic awareness. They tend to treat different lexical variations of the same concept (e.g., synonyms or paraphrases) as unrelated, and may overemphasize surface frequency rather than deeper topical relevance. Consequently, while statistical feature–based methods remain a cornerstone of unsupervised approaches due to their efficiency and interpretability, they are increasingly combined with linguistic or semantic features to achieve more robust performance.

3.3 Graph Based Unsupervised Keyphrase Extraction

Graph-based methods conceptualize a document as a graph, where words (or multi-word phrases) serve as nodes and edges represent their statistical or semantic relationships. The importance of each node is typically estimated using centrality measures, with highly ranked nodes or node sequences selected as candidate keyphrases. Later developments have been made by incorporating the **context extension**, **topic**, **position**, and **embedding** information to enrich the attributes of nodes and edges. We introduce these four aspects below.

1) Context Extension Based Methods. Keyphrase sets from similar texts often overlap, allowing models to leverage these similarities to enhance the original graph's nodes and edges, highlighting important terms. SingleRank [63] extends the traditional TextRank framework by incorporating a small set of nearest-neighbor documents to enrich the context of the target text. Instead of relying solely on local co-occurrence information within a single document, SingleRank constructs a graph over the expanded document set and applies ranking algorithms to identify salient terms. By integrating both document-specific and corpus-level cues, it enhances the robustness of keyphrase extraction and alleviates the sparsity issue of short or narrow texts.

CiteTextRank [22] builds upon the TextRank framework by leveraging citation networks in scientific literature. Instead of limiting the context to the target document, it incorporates vocabulary from cited and citing documents to construct a richer graph representation.

These context extension based methods share the advantage of augmenting the graph with external information, thereby alleviating data sparsity and strengthening the connectivity of nodes. By contextualizing the target document with related texts or citation networks, they can better capture domain-specific terminology and improve the salience of extracted keyphrases. However, their performance is highly dependent on the availability and quality of external resources. Misaligned or noisy neighboring texts may introduce irrelevant information, leading to transfer noise and potentially degrading the extraction quality.

2) Topic Based Methods. Documents generally revolve around a main theme and several sub-themes, each of which can be characterized by different

keyphrases. To avoid redundancy and improve coverage, topic-based methods group candidate phrases into semantically coherent clusters and then select representative keyphrases for each cluster.

The TopicRank model [11] operationalizes this idea by clustering candidate phrases into topics and ranking the clusters, with the top-ranked phrase from each cluster chosen as a keyphrase. This ensures that the extracted set covers diverse aspects of the document rather than overemphasizing frequent but similar terms.

The MultipartiteRank model [7] further enhances this framework by representing keyphrases as nodes in a multipartite graph, where directed edges are only established between candidates from different topics. This structure explicitly enforces inter-topic competition while avoiding intra-topic redundancy, leading to more diverse and balanced keyphrase sets.

The key advantage of topic-based methods lies in their ability to promote diversity and capture clear topic structures, ensuring that extracted keyphrases represent different thematic aspects of the document. Nevertheless, their performance is sensitive to clustering quality and they often struggle to resolve fuzzy or overlapping topics.

3) Position Based Methods. Structural cues such as titles, headings, and sentence boundaries often signal important content, as authors tend to place salient information at the beginning or end of a document or paragraph. Position-based methods explicitly model these positional regularities to improve keyphrase extraction.

The PositionRank model [20] extends the TextRank framework by incorporating positional information into the graph weighting process. Words that appear earlier in the document receive higher importance, under the assumption that they are more likely to reflect the main topics.

The JointGL model [35] generalizes this idea by introducing a boundary function $d_b(i) = \min(i, \alpha(n - i))$, where n is the total number of candidate keyphrases and α is a hyperparameter that balances the contributions of the beginning and ending positions. If $d_b(i) < d_b(j)$, then phrase i is deemed closer to the boundary, and the centrality contribution of j to i is reduced. This ensures that phrases near the document boundaries are emphasized in the ranking process.

Overall, position-based methods are advantageous because they exploit document structure cues that are both interpretable and computationally efficient. However, their reliance on positional heuristics makes them less robust in settings where salient information is not consistently located at document boundaries. In particular, their performance tends to drop on unstructured or noisy layouts, such as informal texts, web content, or domains with irregular writing conventions, which limits their generalizability across genres and domains.

4) Embedding Based Methods Pretrained word embeddings encode rich semantic information that can significantly enhance the quality of graph construction in keyphrase extraction. Unlike co-occurrence statistics, which only reflect surface-level proximity, embeddings capture semantic similarity between terms, thereby enabling more meaningful connections among nodes in the graph.

Table 1: The taxonomy of representative studies on unsupervised keyphrase prediction

Category		Methods	Advantages	Disadvantages
Extraction Methods				
Traditional	Syntactic Feature	PoS-tagging 2003 [25], YAKE 2020[12], HAKE 2022 [41]	Efficient, interpretable, and broadly applicable	Ignores context, structure, and semantic similarity; Limited support for multi-word and synonymous phrases.
	Statistical Feature	TF-IDF 1972 [60] , YAKE 2020 [12], HAKE 2022 [41]	Precise noun-based extraction with rule-compatible filtering.	Poor cross-lingual transfer; sensitive to POS tagging errors.
Graph-based	Context Extension	SingleRank 2008 [63], CiteTextRank 2014 [22]	Contextualized via external data; enhances graph linkage.	Dependent on aligned external texts; prone to transfer noise.
	Topic	TopicRank 2013 [11], MultipartiteRank 2018 [7]	Promotes diversity and clear topic structures	Sensitive to clustering; weak at resolving fuzzy topics.
	Position	PositionRank 2017 [20], JointGL 2021 [35]	Leverages leading sentences and titles to highlight key phrases.	Performance drops on unstructured or noisy layouts
	Embedding	JointGL 2021 [35]	Captures deep semantics and context-dependent meanings.	Resource-heavy and dependent on model and task tuning.
PLM-based	Text Semantics	EmbedRank 2018 [5], SIFRank 2020 [61]; MDERank 2021 [74], HGUKE 2023 [57]	Easily integrates into neural architectures.	Hard to interpret distances in vector space
	Attention Mechanism	AttentionRank 2021 [17], SAMRank 2023 [27]	Leverages attention maps; no extra training	May prefer short phrases; inconsistent granularity
	Prompt Learning	PromptRank 2023 [31], KPEprompt Comparison 2024 [54]; ChatGPT-prompting 2023 [55]	Zero-shot capable; allows prior knowledge injection	Output varies with prompt wording
Generation Methods				
Pseudo Data-based	Datasets-based Pseudo-label Construction	AutoKeyGen 2022 [53], TPG 2024 [28], Silk 2024 [8]	Easy to generate; optimized for specific domains	Domain-sensitive; misses unseen keyphrases
	Corpus-based Pseudo-label Construction	OpenDomainKP 2023 [18], LLM-Improve 2024 [2]	Adaptable across domains with phrase pool expansion	Parsing and embedding quality limits performance
LLM -based	Prompt Based Generation	LLMs Prompt 2024 [56, 29, 43], Long Document Processing 2023 [39],	No labels needed; supports long contexts	May hallucinate; hard to evaluate correctness
	Multi-Stage Generation	LLM-TAKE2023 [38], Thinking2024[65]	Mimics human reasoning process, improves generation quality	Long reasoning chains, lower efficiency
	Post-Process Mechanisms	Output Correction 2024 [68],One2Set+LLM 2024 [51]	Reduce semantic redundancy and noise (semantic deduplication, frequency filtering)	Introduce additional computational overhead

Note: PLM = Pretrained Language Model, LLM = Large Language Model.

This semantic enrichment helps identify keyphrases that may not frequently co-occur but are conceptually related.

The JointGL model [35] illustrates this approach by using BERT [16] to encode candidate phrases into dense vector representations, with edge weights computed via the dot product of embeddings. This embedding-based similarity measure replaces raw co-occurrence counts, allowing the graph to reflect deeper semantic relationships. With the rapid advancement of pretrained language models (PLMs), particularly contextualized embeddings from transformers such as BERT, embedding-driven graph construction has become a mainstream paradigm in unsupervised keyphrase extraction. These methods not only improve semantic coverage but also generalize more effectively across domains compared to traditional surface-level heuristics.

Embedding-based methods offer notable advantages, including their ability to capture deep semantic relations and leverage transfer learning from large-scale corpora, which boosts performance even in low-resource domains. However, they also come with challenges: embedding models are often computationally expensive, resource-heavy and dependent on model/task tuning, may inherit biases or outdated knowledge from their pretraining data, and their reliance on pretrained models can reduce interpretability compared to simpler statistical or positional heuristics.

3.4 Pretrained Language Model Based Unsupervised Keyphrase Extraction

Pretrained language models (PLMs) encode rich linguistic knowledge and contextual semantics learned from large-scale corpora, providing a powerful foundation for keyphrase extraction. Unlike traditional statistical or graph-based methods that primarily rely on surface-level co-occurrence or syntactic cues, PLMs enable models to capture nuanced semantic relationships, contextual dependencies, and domain-transferable representations. Consequently, PLM-based approaches have emerged as the state-of-the-art in unsupervised keyphrase extraction, significantly advancing both robustness and generalization. These approaches can be broadly categorized into three techniques: **text semantics computation**, **attention mechanisms**, and **prompt learning**.

Text Semantics Based Methods. Pretrained language models encode textual semantics into low-dimensional vector representations, which can be leveraged to compute similarity between candidate keyphrases and the source document. The EmbedRank model [5] is a representative approach, using PLMs to encode both the document and candidate phrases and ranking keyphrases based on cosine similarity.

With continuous optimization of pretrained language models, semantic representation has been significantly improved. SIFRank [61] enhances the original EmbedRank embeddings by combining a sentence embedding model (SIF) with the autoregressive pre-trained language model ELMo. This integration improves

the quality of keyphrase ranking, particularly for short documents. SIFRank also introduces optimization techniques such as document segmentation and contextual word embedding alignment to accelerate computation while maintaining high accuracy.

Subsequent methods have further refined this paradigm from multiple perspectives, including handling long documents, modeling hierarchical semantic structures, and adopting task-specific pretraining strategies.

1) Long Document Processing. When processing long documents, the significant difference in sequence lengths between candidate keyphrases and the full text can reduce the accuracy of embedding similarity calculations. To address this challenge, MDERank [74] introduces a masked document strategy: candidate keyphrases are replaced with a [MASK] token in the source document to construct a modified version, and the embedding similarity between this masked document and the original document is computed. This approach indirectly ranks candidate keyphrases while mitigating the bias introduced by long sequences.

Similarly, the HGUKE model [57] tackles long-text issues by selecting a representative subset of the document as a proxy for the entire text. By focusing on a smaller, informative segment rather than the full document, HGUKE reduces the influence of general or repetitive content and emphasizes key information, improving the accuracy and efficiency of keyphrase ranking in lengthy documents.

2) Hierarchical Semantic Modeling. Directly computing embedding similarity between candidate phrases and the full document may overlook multi-level semantic structures, often producing a homogeneous set of keyphrases. To address this limitation, several hierarchical semantic modeling approaches have been proposed.

The CentralityRank model [59] explicitly computes embeddings at three levels—word, phrase, and document—and ranks candidate keyphrases based on their relevance across all levels. Similarly, the HGRRM model [76] first evaluates sentence importance and then ranks keyphrases within each sentence, effectively incorporating sentence-level context.

The HyperRank model [58] represents hierarchical semantics in hyperbolic space, enabling richer, tree-like structures. By mapping both phrase and document embeddings into a shared hyperbolic space and calculating Poincaré distances, HyperRank captures semantic proximity more effectively than standard Euclidean embeddings.

In contrast, the INSPECT model [26] adopts an implicit approach, capturing topic-level information rather than explicitly modeling multiple semantic levels. It assigns keyphrases to distinct topics within the document, which helps maintain diversity in the keyphrase set while summarizing the main themes.

3) Task-specific Pretraining. To further enhance the quality of keyphrase embeddings, several works design task-specific self-supervised pretraining strategies tailored for keyphrase prediction. These methods go beyond generic pre-trained models by explicitly aligning the representation learning process with the properties of keyphrases.

The KPEBERT model [74] extends BERT through contrastive pretraining. Specifically, it constructs three document views: (1) the original document, (2) the document with important phrases masked, and (3) the document with general phrases masked. By applying contrastive learning across these views, KPEBERT encourages the model to distinguish between keyphrase-bearing contexts and non-informative contexts, thereby improving its ability to represent and rank keyphrases effectively.

The KeyBART model [33] introduces a more comprehensive multi-task pretraining framework. In addition to standard random token masking, it incorporates (a) keyphrase boundary filling, where masked spans correspond to entire keyphrases rather than individual words, and the model learns to recover them, and (b) keyphrase replacement classification, where the model must determine whether a substituted span corresponds to a genuine keyphrase. These tailored objectives explicitly expose the model to the structural and semantic characteristics of keyphrases. As a result, KeyBART significantly improves performance not only in unsupervised keyphrase extraction but also in related tasks such as keyphrase generation and named entity recognition, demonstrating the benefit of targeted pretraining.

Text semantics based methods benefit from pretrained language models, which provide powerful contextual representations and capture semantic relationships beyond surface-level statistics. By ranking candidate phrases according to their semantic similarity with the source text, these methods can effectively identify meaningful keyphrases, including those that are not explicitly repeated in the document. Moreover, with the continuous evolution of stronger embedding models, such as ELMo, BERT, and beyond, semantic-based methods consistently achieve improved extraction quality, particularly in short or well-structured documents.

However, these approaches also face several challenges. First, when applied to long documents, the discrepancy in sequence length between candidate phrases and the entire text can weaken similarity calculations, leading to biased rankings. Second, focusing solely on semantic similarity tends to generate homogeneous keyphrase sets, overlooking diversity and coverage. Third, the quality of results is highly dependent on the choice of pretrained models, making them resource-intensive and sensitive to domain shifts.

Attention Mechanism Based Methods. Attention mechanisms enable unsupervised keyphrase extraction models to assign context-sensitive importance scores to words or phrases within a document, capturing both local and global semantic relevance. By utilizing the fixed key-query-value (KQV) matrices inherent in pretrained language models, these approaches can exploit rich contextual embeddings without requiring task-specific fine-tuning.

The AttentionRank model [17] exemplifies this approach by combining self-attention and cross-attention scores. Specifically, for each candidate phrase c, the self-attention score a_c evaluates its relevance to the sentence it appears in, capturing local context. Simultaneously, the cross-attention score r_c measures the phrase's relevance to the entire document, enhancing global semantic alignment.

The final importance score is obtained as a weighted linear combination of a_c and r_c, enabling a balanced consideration of local and document-level significance. This strategy allows AttentionRank to prioritize phrases that are both contextually salient and globally representative.

Similarly, SAMRank [27] introduces a two-fold attention-based scoring mechanism. It first computes a global attention score by summing the attention weights from all other tokens to a candidate phrase, effectively capturing the phrase's influence within the document. Next, it calculates a proportional attention score, based on the principle that tokens attending strongly to important tokens are themselves likely to be important. The final phrase importance score is the sum of these two components, allowing the model to recognize both intrinsically significant phrases and those highlighted by contextual interactions. SAMRank demonstrates that combining global and proportional attention can improve robustness across domains.

Building on the attention mechanism, leveraging a larger and more advanced language model enables more precise attention score computations, leading to enhanced task performance. Nevertheless, these methods inherently depend on the attention distributions of pretrained models, which are not explicitly optimized for keyphrase extraction and may overemphasize syntactic rather than semantic signals. As a result, attention-based approaches excel in extracting explicit keyphrases with strong interpretability and cross-domain robustness, but they struggle with implicit keyphrase generation and often require additional post-processing to mitigate redundancy and improve coverage.

Prompt Learning Based Methods. Prompt learning utilizes natural language prompts to activate the knowledge embedded in pretrained language models. Task-specific prompt templates can be designed for unsupervised keyphrase extraction, applicable to both lightweight pretrained models and large language models.

The PromptRank model [31] leverages a pretrained sequence-to-sequence model (T5) as the backbone and reformulates keyphrase extraction as a prompt-based relevance estimation task. Specifically, it constructs cloze-style templates such as "Book: [document]" and "This book is mainly about [candidate phrase]." By filling the document and candidate phrase into these templates, both are mapped into a shared latent space, where the semantic similarity between them is reflected by the decoder's probability of generating the candidate phrase c. To further refine ranking, PromptRank introduces a position-based penalty r_c under the assumption that salient information is more likely to occur at the beginning of a document, thereby reducing the score p_c for phrases located in less informative positions.

Building on this line, recent work[54] further explores variations in template design to evaluate candidate importance more effectively. Candidate phrases are first extracted via POS-based patterns and deduplicated, then inserted into decoder templates such as "This article mainly discusses [candidate]" or "The keywords of this article are [candidate]." The generation probability, normalized

by phrase length, serves as the ranking score, and a Top-K strategy selects the final keyphrases. Comparative studies of template variants show that explicit keyword-oriented instructions (e.g., "The keywords of this article are ...") can provide stronger task alignment than generic cloze-style prompts, while still requiring no labeled data.

Prompt learning–based methods adapt PLMs to keyphrase extraction by reformulating the task into prompt-driven language modeling objectives, often combined with lightweight fine-tuning. Their advantages include strong adaptability to new domains with minimal labeled data, effective utilization of PLM prior knowledge for both present and absent keyphrases, and improved interpretability since the prompt explicitly links the task to natural language instructions.

However, their performance is highly sensitive to the design of prompt templates and verbalizers, and suboptimal choices can lead to unstable results across datasets. These methods also inherit biases from pretrained models, sometimes overemphasizing frequent or surface-level patterns rather than domain-specific semantics. In addition, while parameter-efficient, prompt learning often requires careful hyperparameter tuning.

4 Unsupervised Keyphrase Generation Methods

4.1 Pseudo-Data Pretrained Unsupervised Keyphrase Generation

Synthetic pseudo-data generation provides a practical means to alleviate the reliance on costly human annotations. By leveraging either existing labeled datasets to construct noisy variants or external corpora to synthesize training pairs, these approaches enable the pretraining of keyphrase generation models at scale. Such methods effectively enhance model generalization and improve the ability to predict absent keyphrases.

Datasets-based Pseudo-labeled Data Construction. Texts within the same dataset typically belong to a common domain, such as KP20k [40] in computer science, which leads to overlapping keyphrase sets across documents. Building on this property, the AutoKeyGen model [53] constructs pseudo-labels by first aggregating noun phrases from the dataset into a phrase bank, then selecting candidates based on their occurrence in the input document. The ranked candidates are subsequently used as pseudo-labels to train a sequence-to-sequence model.

Nevertheless, directly building a phrase bank overlooks the varying importance of different textual components. To address this, a recent study [28] generates pseudo-labels from document titles, thereby improving absent keyphrase generation by prioritizing title information. Similarly, the Silk model [8] derives pseudo-labels from citation contexts, guided by principles of importance, relevance, and reliability, and demonstrates strong performance across domains such as natural language processing, astrophysics, and paleontology.

Datasets-based pseudo-label construction effectively leverages domain-specific corpora to generate high-quality training signals without manual annotation. By utilizing textual elements such as titles or citation contexts, these methods can prioritize important phrases and improve the generation of absent keyphrases. However, their reliance on dataset-specific cues also limits generalizability: models trained on one domain may perform poorly on others, and frequent or surface-level phrases are often favored at the expense of rare but semantically significant expressions. Furthermore, focusing heavily on certain components, such as titles or citations, may overlook other informative parts of the document, potentially reducing coverage and diversity of the generated keyphrases.

Corpus-based Pseudo-labeled Data Construction. Pseudo-labels obtained from individual datasets are typically restricted to a specific domain, which limits their usefulness for cross-domain training. Generating keyphrases that generalize across diverse domains remains a significant challenge, particularly in scenarios with limited labeled data. In such cases, external corpora provide a promising resource for constructing pseudo-labels.

To enable cross-domain keyphrase generation, the OpenDomainKP model [18] extracts grammatically valid noun phrases along with their contextual information from external corpora to build a comprehensive phrase repository. For each phrase z, its context is encoded as v_z. Similarly, the input text x is encoded as v_x, and the cosine similarity between v_z and v_x is computed as their relevance. The top-k most relevant phrases are then retrieved from the repository, forming a pseudo-label set that incorporates external knowledge for model training. By incorporating external knowledge in this manner, OpenDomainKP is capable of performing unsupervised keyphrase generation across multiple domains with minimal adaptation. Its main advantage lies in its flexibility: expanding the phrase repository allows the model to adapt to new domains without the need for domain-specific training data, opening up opportunities for broader application and further research into leveraging external corpora effectively.

Bai et al.[2] adopt a two-stage framework of generation and distillation for unsupervised keyphrase generation with LLMs. In the first stage, the generation process leverages LLMs' strong zero-shot and few-shot prompting abilities. With zero-shot prompting, models generate keyphrases directly from task descriptions, while few-shot prompting enriches the input with a handful of labeled examples to improve contextual learning. To make such methods more practical for real-world deployment, a second stage of knowledge distillation is introduced. Here, the keyphrases generated by LLMs are used as pseudo-labels to fine-tune smaller, lightweight models such as UniLMv2[3].

Corpus-based pseudo-label construction offers strong flexibility and cross-domain adaptability, enabling models like OpenDomainKP to leverage diverse phrase repositories for unsupervised keyphrase generation across multiple domains with minimal adaptation. Nevertheless, the approach is limited by the coverage and relevance of the external corpus, and generated pseudo-labels may include noisy or less informative phrases, potentially affecting overall precision and utility.

4.2 Large Language Model Based Unsupervised Keyphrase Generation

LLMs with increased parameter sizes have recently exhibited remarkable performance in zero-shot natural language processing tasks [48]. Unlike earlier domain-specific pretrained models (e.g., SciBART [67]), which are costly to train and limited in transferability, LLMs benefit from more diverse corpora and multi-task pretraining, making them highly promising for keyphrase generation. Recent studies have investigated different ways of adapting LLMs for this task, which can be broadly grouped into three categories: (1) prompt-based generation, (2) multi-stage generation, and (3) post-process mechanisms.

Prompt Based Generation. Owing to multi-domain and multi-task pretraining, LLMs are capable of generating keyphrases in an unsupervised manner through prompts, often surpassing state-of-the-art unsupervised models in semantic-based evaluations and approaching the performance of supervised approaches.

Song et al. [56] demonstrate that careful prompt design is crucial for guiding LLMs in zero-shot keyphrase generation. Instructions that use extraction- or generation-oriented verbs can steer the model to focus on present (explicit) or absent (implicit) keyphrases. Enforcing structured output formats reduces noise and improves consistency, while hybrid prompting strategies—combining direct generation with candidate filtering—enhance both accuracy and diversity.

Kang et al. [29] further explore four distinct prompting strategies: vanilla prompting, which leverages the LLM's inherent semantic knowledge to extract keyphrases in a specified format; role prompting, which assigns the model a task-specific role to activate reasoning patterns; candidate-based prompting, which first constructs a candidate pool of noun phrases via POS tagging and lets the LLM select the most relevant ones; and hybrid prompting, which combines the generative flexibility of vanilla prompting with the boundary precision of candidate-based prompting, achieving consistent improvements on large-scale models.

In addition, zero-shot keyphrase generation benefits from prompt-based aggregation. Multiple outputs from different prompts or repeated generations can be consolidated using strategies such as union, interleaving, or frequency-based ranking, improving recall and overall quality without requiring labeled data [43].

A practical advantage of prompt-based LLMs keyphrase generation is its enhanced capability to handle long documents. By designing prompts that guide the model to focus on relevant content throughout the text, LLMs such as ChatGPT can effectively extract keyphrases from extended inputs, significantly improving performance on long-document datasets [39].

Multi-Stage Generation. Beyond single-turn prompting, recent approaches exploit multi-step reasoning to better emulate human keyphrase annotation. Wang et al. [65] propose a four-stage framework—extraction, extension, retrieval,

and ranking—where the extractor identifies candidate phrases directly present in the document, the extender broadens coverage via hypernyms and synonyms, and the retriever incorporates relevant phrases from similar-domain documents to enhance absent keyphrase prediction. A multi-turn ranking step then organizes and filters candidates, producing a refined final set.

Similarly, LLM-TAKE [38] employs a dual-mode prompting strategy, combining extraction-based keyphrases grounded in the text with generation-based keyphrases that leverage contextual reasoning. A three-tier filtering mechanism removes low-frequency, sensitive, or low-confidence candidates, while a dynamic ranking integrates LLMs confidence, reference-set frequency, and semantic deduplication.

These multi-step reasoning and multi-stage generation frameworks demonstrate the advantages of LLMs' zero-shot capability, extended token limits, and rich domain knowledge, allowing models to generate high-quality, semantically coherent, and theme-consistent keyphrases from both short and long documents.

Post-Process Mechanisms. While LLMs can generate keyphrases effectively from zero-shot prompts, they sometimes conflate keyphrase generation with entity extraction or produce overly literal outputs. To address this, several strategies have been proposed. Wu et al. [68] introduce a self-consistency decoding mechanism that exploits frequency-based signals to retain the most informative phrases, improving performance with GPT-3.5-turbo and GPT-4.

The One2Set+LLM framework [51] proposes a generate-then-select paradigm, where ONE2SET acts as a generator using an optimal transport–based assignment strategy to improve candidate recall, while an LLM-based selector reformulates selection as a sequence labeling task. This division of labor balances precision and recall: the generator produces diverse candidates, and the LLM refines them for accuracy.

Similarly, Zanutto et al. [73] combine LLM-generated candidates from zero-shot and few-shot prompts with a post-processing refinement module. Semantic redundancy is reduced via sentence embedding similarity, and low-frequency noise is filtered by analyzing phrase occurrences across similar reviews. These approaches collectively demonstrate that combining generation with post-processing can substantially enhance keyphrase quality.

Overall, LLM-based keyphrase generation methods exhibit several notable advantages. They leverage extensive pretraining across multiple domains and tasks, enabling zero-shot generation of both present and absent keyphrases without reliance on labeled data. Carefully designed prompts can guide the model to focus on specific types of keyphrases, enforce structured outputs, and balance diversity and precision. Multi-stage strategies further enhance coverage, while post-processing mechanisms such as semantic deduplication and frequency-based filtering improve quality and reduce noise. Moreover, the large context windows of modern LLMs allow effective processing of long documents, a limitation for many traditional approaches.

However, these methods also have certain limitations. Performance heavily depends on the quality of prompt design. LLMs may still produce hallucinated or irrelevant keyphrases, particularly in specialized domains, and post-processing steps are required to mitigate this. Finally, while zero-shot methods are effective, they may underperform compared to domain-tuned or task-specific supervised models in scenarios with abundant labeled data.

5 Keyprase Evaluation Metrics

Keyphrase prediction results can be evaluated in three ways: reference-based metrics, reference-free metrics, and human evaluation. Reference-based metrics compare the predicted keyphrase set with a reference set (label data), requiring high-quality labels. Reference-free metrics do not rely on labels but need to be task-specifically designed. Human evaluation reflects human judgment but requires careful design to ensure consistency and reproducibility. Below, we describe these three types of evaluation metrics.

	Reference-based				Reference-free			Human-eval
	F1-Score	SoftKey Score	SemR-P	ParaScore	KPEval	ParaScore-free	CAME	Human
Coverage	✓	✓	✓	✓	✓	✓	✓	✓
Distinctiveness	✗	✗	✗	✗	✗	✗	✓	✓
Conciseness	✗	✗	✗	✗	✗	✗	✓	✓
Diversity	✗	✗	✗	✓	✓	✓	✓	✓

Fig. 4: Analysis of Evaluation Metrics from a Linguistic Perspective

5.1 Reference-based Evaluation

A high-quality reference set serves as a representative summary of the key information in the source document. Reference-based evaluation methods measure model performance by quantifying the alignment between predicted and reference keyphrase sets, typically through either lexical or semantic matching.

1) Lexical Matching Classical metrics include Precision, Recall, and F1-score[47], which evaluate phrase-level lexical overlap between predictions and references (Eq.1). Here, TP denotes correctly identified keyphrases, FP refers to non-keyphrases incorrectly predicted as keyphrases, and FN represents keyphrases missed by the model. In practice, $Recall@k$ and $F1@k$ are widely adopted, where k specifies the number of top-ranked predictions considered.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

Based on the fundamental metrics, Yuan et al. [72] noted that the value of k in $F1@k$ is typically fixed at 5, 10,or 15. However, the number of keyphrases in

Table 2: Performance of various keyphrase prediction methods across six benchmark datasets.

Category	Method	Inspec		SemEval-2010		KP20k		Krapivin2029		DUC2001		NUS	
		F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10
Present Keyphrase													
Statistical	TF-IDF [60]	0.112	0.138	0.028	0.034	0.072	0.094	0.115	0.150	0.092	0.106	0.116	0.142
	YAKE [12]	0.181	0.196	0.117	0.144	0.093	0.171	0.122	0.143	0.122	0.143	-	-
Graph-based	JointGL [35]	0.326	0.402	0.130	0.193	-	-	-	-	-	-	-	-
	TextRank [42]	0.270	0.251	0.038	0.053	0.181	0.151	0.094	0.121	0.286	0.355	-	-
	SingleRank [63]	0.277	0.344	0.059	0.090	0.099	0.124	0.105	0.097	0.118	0.182	0.018	0.030
	TopicRank [11]	0.253	0.284	0.121	0.129	0.090	0.138	0.204	0.255	0.029	0.045	-	-
	PositionRank [20]	0.281	0.328	0.098	0.133	0.148	0.145	0.215	0.231	0.045	0.079	-	-
	MultipartiteRank [7]	0.259	0.295	0.121	0.137	0.143	0.138	0.093	0.190	0.233	0.285	0.061	0.085
	MDERank	0.278	0.343	0.130	0.182	0.123	0.143	0.233	0.266	0.143	0.184	-	-
PLM-based	HyperRank [58]	0.333	0.407	0.147	0.213	-	-	-	-	-	-	-	-
	SAMRank [27]	0.339	0.393	0.152	0.183	0.147	0.138	0.151	0.140	0.161	0.167	0.172	0.201
	PromptRank [31]	0.317	0.378	0.172	0.206	-	-	0.273	0.315	-	-	-	-
	KeyBART [33]	0.244	-	0.274	-	0.307	-	0.292	-	0.347	-	-	-
	TPG [28]	0.344	-	0.286	-	0.304	-	0.294	-	0.358	-	-	-
PseudoData-based	OpenDomainKP [18]	0.233	-	0.222	-	0.214	-	0.298	-	-	-	-	-
	AutoKeyGen [53]	0.303	0.345	0.187	0.240	0.234	0.246	0.171	0.155	0.218	0.233	-	-
LLM-based	LLM-TLA [65]	0.413	-	0.256	-	0.253	-	0.291	-	-	-	-	-
	ChatGPT [56]	0.401	-	0.262	-	0.179	-	0.230	-	0.287	-	-	-
	One2Set+LLM [51]	0.357	-	0.405	-	0.453	-	0.435	-	-	-	0.528	-
Absent Keyphrase													
PseudoData-based	TPG [28]	0.0277	-	0.0308	-	0.029	-	0.038	-	0.047	-	-	-
	OpenDomainKP [18]	0.021	0.032	0.014	0.023	0.045	0.072	0.018	0.031	-	-	-	-
LLM-based	LLM-TLA [65]	0.004	-	0.003	-	0.005	-	0.002	-	-	-	-	-
	ChatGPT [56]	0.029	-	0.005	-	0.041	-	0.005	-	0.009	-	-	-
	One2Set+LLM [51]	0.064	-	0.058	-	0.112	-	0.126	-	-	-	0.122	-

*Note: Baseline model results are taken from original papers or other studies [31, 33, 40, 58], and ChatGPT's results are from [56].

both the predicted set and the label set is not fixed. However, since the number of keyphrases in both predicted and reference sets is not fixed, they proposed two alternative metrics: $F1@O$ and $F1@M$, where O corresponds to the size of the reference set and M to the size of the prediction set. While widely used, F1-based metrics remain limited in capturing semantic equivalence, motivating the development of semantic matching approaches.

2) Semantic Matching SoftKeyScores [34] extends evaluation beyond surface-level overlap by introducing two complementary metrics: Keyphrase Match Rate (KMR) and Score. KMR adapts the Translation Edit Rate (TER) to measure edit distance between sets of keyphrases, while the Score metric follows a BERTScore-style approach [75], computing greedy embedding-based similarity between predicted and reference phrases. The final evaluation, F_{score}, is the harmonic mean of P_{score} and R_{score} (Eq. 2).

$$P_{\text{score}} = \frac{1}{|G|} \cdot \sum_{g_i \in G} \max_{l_j \in L} \text{score}\left(g_i, l_j\right), \quad R_{\text{score}} = \frac{1}{|L|} \cdot \sum_{l_j \in L} \max_{g_i \in G} \text{score}\left(g_i, l_j\right) \tag{2}$$

KPEval [69] further broadens evaluation by incorporating four dimensions: reference agreement, faithfulness, diversity, and utility. Among them, reference agreement functions similarly to SoftKeyScores, leveraging cosine similarity between embeddings to assess semantic alignment.

Despite their utility, both lexical and semantic matching metrics have limitations. Traditional string-based measures fail to capture semantically equivalent but lexically distinct expressions (e.g., "neural networks" vs. "deep learning"), while most semantic metrics remain insensitive to ranking. To address these issues, Semantic R-Precision (SemR-p) [62] has been proposed as a more comprehensive evaluation framework.

SemR-p is grounded in three principles: (1) *semantic sensitivity*, which accounts for synonyms and semantically related phrases beyond lexical overlap; (2) *ranking sensitivity*, which gives greater weight to higher-ranked predictions to better reflect user attention patterns; and (3) *adaptive evaluation scope*, which adjusts the number of evaluated predictions to match the number of reference phrases R, thereby avoiding arbitrary cutoffs (e.g., top-5 or top-10). The evaluation proceeds as follows: first, only the top-R predictions are considered; second, each prediction is scored—receiving 1.0 if it exactly matches a stemmed reference phrase, or otherwise obtaining a similarity score computed via sentence embeddings against the top-k most similar references; finally, the SemR-p score is calculated as the average across all R evaluated predictions.

In addition, ParaScore [52] has been introduced as a novel evaluation framework that jointly considers semantic fidelity and lexical diversity. It is formally defined as:

$$\text{ParaScore} = \max(\text{Sim}(X, C), \text{Sim}(R, C)) + \omega \cdot \text{DS}(X, C) \tag{3}$$

where X denotes the input document, R the reference paraphrase, C the candidate output, and ω a tunable weight (empirically set to 0.35). The similarity component, based on BERTScore, captures semantic preservation, while the diversity score

$\mathrm{DS}(X, C)$ quantifies lexical variation. The use of $\max(\mathrm{Sim}(X, C), \mathrm{Sim}(R, C))$ ensures robustness across both reference-dependent and reference-free scenarios.

The lexical diversity component employs a piecewise scoring function:

$$DS(X, C) = \begin{cases} \gamma & , \mathrm{d} > \gamma \\ \mathrm{d} \cdot \frac{\gamma+1}{\gamma} - 1 & , 0 \leq \mathrm{d} \leq \gamma \end{cases} \tag{4}$$

where $d = Dist(X, C)$ represents a normalized distance metric, and γ is a hyper-parameter. This formulation penalizes trivial copying ($d = 0$), rewards moderate lexical variation, and caps the effect of excessive differences once $d > \gamma$.

5.2 Reference-Free Evaluation

Labels in benchmark datasets are usually annotated either by authors or by domain experts. Authors, who possess an in-depth understanding of their own work, often provide absent keyphrases that concisely summarize the document. In contrast, experts—depending on their specific background—may emphasize more detailed or peripheral aspects of the text. Consequently, the resulting label sets can be highly subjective, making evaluations that rely exclusively on references potentially unreliable. To address this issue, reference-free evaluation metrics have been developed to provide more robust and context-sensitive assessments.

KPEval[69] incorporates three reference-free metrics. First, Faithfulness measures whether predicted keyphrases are semantically grounded in the source document. Absent keyphrases are considered faithful if they correspond to synonyms, hypernyms, or hyponyms of document concepts, while present keyphrases are judged faithful if their boundaries are precisely identified (e.g., extracting "NP-hard problem" rather than simply "hard problem"). Second, Diversity evaluates whether the predicted keyphrases cover a broad semantic space with minimal redundancy. It is quantified by combining the lexical repetition percentage with the average semantic similarity between phrases. Third, Utility assesses the contribution of keyphrase sets to downstream information retrieval tasks.

Despite their utility, these metrics have limitations. Faithfulness primarily captures local semantic alignment between keyphrases and text, but overlooks global semantic coverage of the document. Furthermore, measuring utility through downstream performance not only incurs high computational cost but also reduces the general applicability of the metric.

ParaScore[52] further extends its framework with a reference-free variant, specifically designed for cases where gold-standard references are unavailable. In this setting, the $\max(\cdot)$ operation in the original formulation is replaced by $\mathrm{Sim}(X, C)$, which relies solely on the semantic similarity between the candidate output and the input document. The formulation is given as:

$$\text{ParaScore.Free} = \mathrm{Sim}(X, C) + \omega \cdot \mathrm{DS}(X, C) \tag{5}$$

where the lexical diversity component $\mathrm{DS}(X, C)$ remains unchanged, ensuring meaningful variation while discouraging trivial copying.

5.3 Human Evaluation

Given the inherent limitations of automatic metrics, human evaluation is often employed to assess the quality of predicted keyphrases. Traditionally, it has been regarded as the de facto gold standard for evaluating experimental results, with its reliability rarely questioned. However, recent findings challenge this assumption. Belz et al. [4] conducted a large-scale review of human evaluation experiments reported in NLP papers over the past five years and found that their reproducibility was as low as 5%. Even when original authors provided direct assistance, reproducibility only increased to 20%.

These results highlight a critical challenge: in the absence of universally accepted guidelines for human evaluation in keyphrase prediction, achieving high inter-annotator consistency and ensuring reproducibility remain difficult. Consequently, while human evaluation continues to play a crucial role, its subjectivity and methodological inconsistencies limit its reliability as a standalone evaluation framework.

The comparative performance of representative keyphrase prediction methods across benchmark datasets is summarized in Tab. 2.

6 Context-Aware Multi-dimensional Evaluation

6.1 CAME-base

Based on the analysis of keyphrase properties in Sec.2, we propose a context-aware evaluation framework CAME (**C**ontext-**A**ware **M**ulti-dimensional **E**valuation for keyphrase generation). CAME provides a comprehensive assessment of model performance across multiple dimensions, while significantly reducing reliance on labeled data during evaluation.

Despite recent progress in reference-free evaluation, existing metrics remain limited in scope. Most focus on isolated aspects, such as coverage or diversity, but fail to jointly capture the four core properties of keyphrases. This gap limits the capacity of existing metrics to deliver holistic and reliable evaluation. To address these shortcomings, we propose CAME, a unified context-aware framework that integrates complementary perspectives for a more comprehensive assessment. The framework of CAME is illustrated in Fig. 5.

Coverage evaluates the relationship between the generated keyphrase set and the input text, measuring how comprehensively the keyphrases summarize the document content. A keyphrase set that fails to capture important aspects of the document should be penalized, whereas one that effectively covers the core themes of the text should be rewarded.

Existing evaluation metrics often treat the reference keyphrase set as a proxy for the original document, and assess coverage by measuring lexical overlap or semantic similarity between the generated and reference sets—such as using F1-score or SoftKeyScores. However, these metrics are sensitive to the quality of the reference set and may not reliably reflect performance in real-world applications.

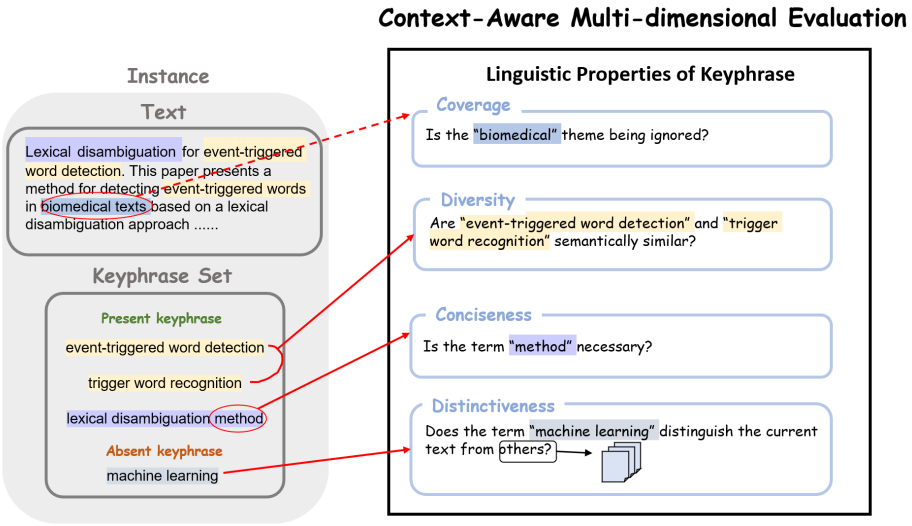


Fig. 5: The framework of CAME.

To address this limitation, we propose measuring coverage directly with respect to the input document, treating the document as contextual background. To capture multi-level topics in the text, we encode the input at both the document and sentence levels to form a set of contextual embeddings S, which incorporates global and local semantic information, respectively. The document-level embedding is derived from the special [CLS] token, while sentence-level embeddings are computed as the average embeddings of all tokens in each sentence.

Recognizing that different parts of the text contribute unequally to its meaning, we retain the self-attention maps from the pretrained language model during encoding to capture the relative importance of different segments. Meanwhile, the generated keyphrase set is also encoded using the same pretrained model to obtain a set of embeddings G. Based on these embeddings, we compute the coverage of the generated set with respect to the contextual embeddings, as defined in Equation 6.

$$P_{\text{Coverage}} = \frac{1}{|S|} \sum_{g_i \in G} \max_{s_j \in S} \frac{s_j^\top \hat{g}_i}{\|s_j\| \, \|\hat{g}_i\|} \quad R_{\text{Coverage}} = \frac{1}{|G|} \sum_{s_j \in S} \max_{g_i \in G} \frac{s_j^\top \hat{g}_i}{\|s_j\| \, \|\hat{g}_i\|} \tag{6}$$

$$\text{Cov} = 2 \cdot \frac{P_{\text{Coverage}} \times R_{\text{Coverage}}}{P_{\text{Coverage}} + R_{\text{Coverage}}} \tag{7}$$

Distinctiveness evaluates whether the keyphrases in a set capture domain-specific knowledge rather than merely reflecting general, cross-domain concepts. A keyphrase set with high distinctiveness helps distinguish the target document from others. This property can be assessed from two perspectives: domain specificity

and text representativeness. Domain specificity is quantified using Pointwise Mutual Information (PMI), which measures the statistical association between keyphrases and domain-specific terms. Text representativeness is quantified using TF-IDF, which reflects the importance of keyphrases within the document relative to a larger corpus.

For a keyphrase $K = (w_1, w_2, ..., w_m)$ consisting of m words, PMI is used to quantify the co-occurrence likelihood of adjacent word pairs relative to their independent occurrences. This reflects the collocational stability of word sequences, thereby serving as an indicator of the domain specificity of the phrase. The PMI score for a bigram (w_i, w_{i+1}) is computed as:

$$\mathrm{PMI}\left(w_i, w_{i+1}\right) = \log \frac{\mathrm{count}\left(w_i, w_{i+1}\right)}{\mathrm{count}\left(w_i\right)\mathrm{count}\left(w_{i+1}\right)} \tag{8}$$

The PMI score of a multi-word phrase K is defined as the average PMI over all adjacent word pairs within the phrase:

$$\mathrm{PMI}(K) = \frac{1}{m-1}\sum_{i=1}^{m-1}\mathrm{PMI}\left(w_i, w_{i+1}\right) \tag{9}$$

TF-IDF evaluates the representativeness of a phrase by measuring how frequently it appears in a given document while penalizing its frequency across the entire corpus. A higher TF-IDF score indicates that the phrase occurs frequently in the current document but rarely across other documents, thus capturing its document-level salience.

The TF-IDF score of a phrase K in document d with respect to a corpus D is computed as follows:

$$\mathrm{TF}(K, d) = \frac{f_{K,d}}{\sum_{w' \in d} f_{w',d}} \tag{10}$$

$$\mathrm{IDF}(K, D) = \log \frac{|D|}{|\{d' \in D : K \in d'\}|} \tag{11}$$

$$\text{TF-IDF}(K, d, D) = \mathrm{TF}(K, d) \cdot \mathrm{IDF}(K, D) \tag{12}$$

Where $f_{K,d}$ denotes the frequency of phrase K in document d, w' is any word in d, $|D|$ is the total number of documents in the corpus, and the denominator of Equation 11 counts how many documents in D contain phrase K.

Finally, the overall Distinctiveness score of a keyphrase K is computed by linearly combining its domain specificity (PMI) and document representativeness (TF-IDF):

$$\mathrm{Dis}(K) = \lambda \cdot \mathrm{PMI}(K) + (1-\lambda) \cdot \text{TF-IDF}(K, d, D) \tag{13}$$

Where $\lambda \in [0, 1]$ is a weighting parameter that balances the contributions of PMI and TF-IDF. The overall distinctiveness score for a generated keyphrase set is obtained by averaging the distinctiveness scores of all keyphrases.

Conciseness emphasizes the clarity of phrase boundaries and brevity of expression, ensuring that each keyphrase accurately captures core concepts without

redundant modifiers or omission of essential information. Evaluation of conciseness considers both semantic completeness and contextual appropriateness. For instance, "neural network architecture optimization" specifies both the technical domain (neural networks) and the optimization target (architecture), forming an inseparable cognitive unit. Removing any component (e.g., architecture) would result in semantic generalization or ambiguity, indicating that the current structure is already minimal and complete.

To assess the conciseness of keyphrases, we define two guiding principles: **1)** The less information gain obtained by extending a phrase with additional words, the more concise the phrase is. **2)** The greater the information loss when truncating a phrase by removing internal words, the more concise the phrase is. By comparing the information content of a candidate keyphrase with that of its extended or reduced variants, we can quantify how effectively the phrase captures essential semantics in a minimal form.

To obtain the extended and sub-phrases, we first extract noun phrases from the original text using a phrase mining tool and identify phrases that have a containment relationship with the generated keyphrase K, denoted as Chunk_K. The information content of a phrase is estimated by its semantic similarity to the input document d. The information difference is computed as follows:

$$\text{Con}(K) = \log_2 \frac{\text{sim}(K, d)}{\text{sim}\left(\text{Chunk}_K, d\right)} \tag{14}$$

Here, the function $\text{sim}(\cdot)$ denotes the similarity computation, which is implemented as the cosine similarity between the two embedding representations.

Diversity measures the semantic independence among keyphrases within a generated set. A diverse keyphrase set avoids redundancy and overlapping meanings, thereby improving the breadth of conveyed information. In prior work, diversity is commonly evaluated at both the lexical and semantic levels. At the lexical level, we first apply stemming to eliminate the influence of morphological variations. Then, we compute the proportion of repeated words across the keyphrase set. A high repetition rate indicates poor diversity in the generated keyphrases. At the semantic level, we measure pairwise cosine similarity between the embeddings of all generated keyphrases. This approach captures the semantic redundancy within the set. For example, the Emb_sim metric in KPEval [69] follows this principle. The semantic diversity score is defined as:

$$\text{Div}(G) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{1}(i \neq j) \, \text{sim}\left(g_i, g_j\right)}{m(m-1)} \tag{15}$$

where $G = g_1, g_2, ..., g_m$ denotes the embedding set of generated keyphrases, and $\text{sim}\left(g_i, g_j\right)$ represents the cosine similarity between two embeddings. The indicator function $\mathbb{1}(i \neq j)$ ensures that only distinct phrase pairs are considered. A lower value of $\text{Div}(G)$ indicates higher diversity, as it implies less semantic overlap among the keyphrases.

6.2 CAME-LLM

LLMs have demonstrated a high degree of consistency with expert evaluations in various natural language generation tasks. Therefore, we leverage LLMs as the backbone of CAME, which integrates four key metrics: coverage, distinctiveness, conciseness, and diversity.

For ***coverage*** evaluation, LLMs are capable of capturing the relevance between keyphrases and textual content, while also understanding multi-level thematic structures within the text. This allows them to assess whether the generated keyphrases adequately reflect both the global theme and important subtopics of the input document, thereby avoiding omissions of critical aspects.

In terms of ***distinctiveness***, LLMs are trained on corpora spanning diverse domains and topics, enabling them to learn domain-specific linguistic patterns and knowledge representations. This domain awareness allows LLMs to identify specialized language features and knowledge distributions, making them suitable for assessing whether the generated phrases exhibit domain-relevant uniqueness—especially when dealing with technical or specialized documents.

Regarding ***conciseness***, LLMs possess strong semantic understanding and contextual modeling capabilities, which facilitate evaluation from two perspectives. First, based on semantic unit completeness, LLMs leverage their linguistic knowledge to determine whether a keyphrase forms a coherent and complete semantic unit, free from fragmented or redundant components. Second, through contextual information optimization, the model assesses whether a keyphrase achieves maximal information density under its specific contextual constraints—ensuring neither excessive modification nor critical omissions.

For ***diversity*** evaluation, LLMs can simultaneously consider lexical and semantic variations across the keyphrase set. This enables the detection of redundant or semantically overlapping keyphrases and ensures that the generated set offers broad and non-redundant informational content.

Chen et al. [13] demonstrated that when using LLMs such as text-davinci-001 for evaluation tasks, prompting the model to explicitly generate numerical scores yields more reliable results than using implicit generation probabilities as scores. Prior work on LLM-based evaluation of text generation quality [36] has also pointed out a common issue: the output scores often exhibit mode collapse—a dominant score (e.g., 3 in a 1–5 range) tends to appear most frequently. This leads to low score variance and frequent ties, which undermines the ability to distinguish subtle differences between generated texts. To address this, we design a prompt-based evaluation mechanism tailored for keyphrase generation. Specifically, we predefine a set of candidate scores as a real-valued set $S = s_1, s_2, ..., s_n$, and let $p(s_i)$ denote the LLM's predicted probability for each explicit score s_i. To improve score precision and capture finer-grained differences, we compute the final evaluation score as a weighted average over all candidates, where each score is weighted by its generation probability:

$$\text{score} = \sum_{i=1}^{n} p(s_i) \times s_i \tag{16}$$

This probabilistic scoring strategy enables more continuous and discriminative evaluation outcomes.

To enable standardized evaluation, we design a unified prompt template. The template consists of five parts:

- **Task Description** – a clear statement of the evaluation task;
- **Score Definition** – an explicit explanation of the scoring scale;
- **Input Slots** – placeholders for the evaluation metric, source text, and the generated keyphrase set;
- **Chain-of-Thought Guidance** – a reasoning structure to guide the large language model toward a well-grounded judgment;
- **Output Prompt** – a direct instruction for the model to produce its score.

In the fourth part, we incorporate a reasoning procedure inspired by expert assessment protocols [14], which includes the following three steps: 1) outlining the evaluation process, 2) detailing the scoring criteria and explaining the metric, 3) requiring the model to justify its assigned score to enhance evaluation consistency. By embedding this expert-style reasoning chain into the prompt, the large language model is guided to conduct structured and rigorous scoring. The complete prompt template is illustrated in Tab. 3, and natural language formulations for each evaluation metric used to fill in the template are shown in Figure 6.

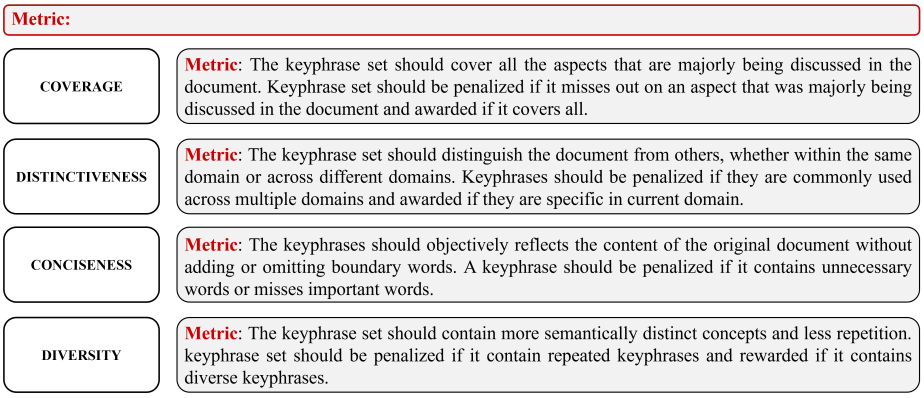


Fig. 6: The description of the evaluation metrics filled in the prompt template

6.3 Datasets

The datasets for the keyphrase prediction task cover various domains such as computer science, news, and biomedicine, with annotations made by authors, readers, or experts. We also investigate emerging cross-lingual and multimodal keyphrase prediction datasets. Tab.4 provides detailed statistical information

Table 3: Unified prompt template for CAME

Task Description: You will be given a document using which a keyphrase set has been generated. Your task is to evaluate the keyphrase set based on the given metric. Evaluate to which extent the keyphrase set follows the given metric considering the document as the input. Use the following evaluation criteria to judge the extent to which the metric is followed. Make sure you understand the task and the following evaluation metric very clearly.
Evaluation Criteria: The task is to judge the extent to which the metric is followed by the keyphrase set. Following are the scores and the evaluation criteria according to which scores must be assigned. `<score>1</score>` - The metric is not followed at all while generating the keyphrase set from the document. `<score>2</score>` - The metric is followed only to a limited extent while generating the keyphrase set from the document. `<score>3</score>` - The metric is followed to a good extent while generating the keyphrase set from the document. `<score>4</score>` - The metric is followed mostly while generating the keyphrase set from the document. `<score>5</score>` - The metric is followed completely while generating the keyphrase set from the document.
Metric: {} **Document: {}** **Keyphrase Set: {}**
Evaluation Steps: Follow the following steps strictly while giving the response: 1. Read the document carefully and check if the keyphrase set adheres to the metric considering the document as the input. 2. Next, evaluate the extent to which the metric is followed. Rate the keyphrase set using the evaluation criteria and assign a score within the `<score>...</score>` tags. 3. Write down the reason why this score is assigned within the `<reason>...</reason>`, focusing on the shortcomings of the keyphrase set. **Note**: Strictly give the score within `<score>...</score>` tags only, e.g., `<score>5</score>`. THE SCORE AND REASON MUST BE ASSIGNED STRICTLY ACCORDING TO THE METRIC ONLY AND NOTHING ELSE!
Response:

about these datasets, categorized by short-text datasets, long-text datasets, and multi-modal/multi-lingual datasets, arranged chronologically.

Table 4: Statistics of datasets

Dataset	Domain	Counts	Length	Annotator	Year
Short-text Dataset(length<=500)					
INSPEC[25]	Comp.Science	2000	128	E	2003
SemEval-2017[1]	Science	500	178	E	2017
KP20k[40]	Comp.Science	568k	176	A	2017
STACKEX[72]	Comp.Science	331k	300	A	2019
KP-Biomd[24]	Biomedical	5.9M	271	A	2022
Long-text Dataset(length>500)					
NUS[45]	Science	211	7644	A&R	2007
DUC2001[63]	News	308	740	R	2008
Krapivin2009[32]	Comp.Science	2304	8040	A	2009
SemEval-2010[30]	Multi-domain	244	7961	A&R	2010
OpenKP[71]	Multi-domain	148k	900	E	2019
KPTimes[21]	News	280k	921	E	2019
LDKP3K[37]	Science	100K	6027	A	2021
LDKP10K[37]	Science	1.3M	4384	A	2021
METAKP[68]	Multi-domain	7500	Mixed	GPT-4&R	2024
Multi-modal/Multi-lingual Dataset					
Tweet-KP[66]	Multi-modal	53781	27	A	2020
Papyrus[46]	Multi-lingual	16427	290-573	A	2022
EUROPA[50]	Multi-lingual	285k	5220	A	2024

*Note: A for authors; R for Readers; E for Experts

We conduct experiments on the three most widely used benchmark datasets for keyphrase generation—**KP20k**, **INSPEC**, and **KP-biomed**—to validate the effectiveness of the proposed CAME evaluation framework.

1. **KP20k** [40] is one of the most commonly used datasets in this task. It contains the titles and abstracts of scientific papers in the computer science domain, with author-assigned keyphrases as ground truth.
2. **INSPEC** [25] consists of abstracts from journal papers in the field of computer science. The keyphrases are manually annotated by domain experts.
3. **KP-biomed** [24] is a biomedical-domain dataset for keyphrase generation, with data collected from PubMed. The keyphrases are provided by the original paper authors.

6.4 Baselines

We evaluate six unsupervised keyphrase generation models, which can be categorized into three types: graph based, pretrained language model based, and LLM based methods.

1. PositionRank [20]: This model incorporates word position and frequency information. It aggregates the inverse of each occurrence position of a word and normalizes the result. The final scores are integrated into the PageRank

algorithm, favoring words that appear early and frequently in the document for keyphrase extraction.
2. MultipartiteRank [7]: Candidate phrases are first clustered into topics, forming a multipartite graph where each node represents a phrase. Nodes from different topics are connected with weighted edges based on their distance, while phrases within the same topic are not connected. The TextRank algorithm is then used to select the most representative phrases from each topic to form the keyphrase set.
3. SAMRank [27]: This model decomposes phrase importance scoring into global attention and proportion-based attention. It leverages the self-attention matrix from a pretrained language model (e.g., GPT-2) to compute attention scores for phrases, which are then ranked to form the keyphrase set.
4. PromptRank [31]: This method uses a pretrained language model, T5 [49], as the backbone. It constructs prompt templates such as "Book: [document]" and "This book is mainly about [candidate phrase]". The generation probabilities of candidate phrases are used for ranking and selecting keyphrases.
5. GPT-3.5-turbo: Optimized for conversational tasks and general-purpose text generation, this model has 175 billion parameters. It improves language understanding, contextual coherence, and multi-turn interaction capabilities through enhanced training data and algorithmic advancements. In this study, it is used in a zero-shot setting with natural language prompts to generate keyphrases.
6. LLaMA-3-Instruct-8B [23]: Designed for better dialogue and complex task execution, this model has 8 billion parameters and adopts a Transformer-based autoregressive architecture. It supports long-context processing up to 128k tokens. In this study, it is also used in a zero-shot setting with natural language prompts for keyphrase generation.

6.5 Experimental Setting

In the CAME-base configuration, we adopt BERT-base-uncased as the pretrained language model. The weighting parameter λ in the distinctiveness metric is set to 0.5 to balance domain specificity (PMI) and document representativeness (TF-IDF).

For CAME-LLM, each variant is named according to its underlying backbone model (e.g., CAME-GPT-3.5-turbo). To evaluate the framework's performance across different large language models, we conduct comparative experiments using GPT-4o, the Qwen2.5 series, and DeepSeek-V3 as backbones. GPT-4o demonstrates stronger complex reasoning capabilities compared to GPT-3.5-turbo. The Qwen2.5 series, developed by Alibaba Cloud, represents a recent family of LLMs; we include both Qwen2.5-32B-Instruct and Qwen2.5-72B-Instruct in our experiments. DeepSeek-V3 is a 671-billion-parameter mixture-of-experts model pretrained on 14.8 trillion high-quality tokens and further refined using supervised fine-tuning (SFT) and reinforcement learning (RL). It achieves state-of-the-art performance among open-source models and approaches the capabilities of leading proprietary systems.

As the GPT models are proprietary and do not expose token-level generation probabilities, we ensure evaluation stability by repeating each scoring prompt five times per input. We approximate the probability distribution over discrete scores by computing their empirical frequency across these repetitions. The final metric score is then computed as a weighted average, using the estimated probabilities.

Currently, there is no benchmark evaluation dataset for multidimensional keyphrase generation, so we design a human evaluation experiment in this section to verify whether CAME aligns with expert preferences. Eight experts in computer science were asked to perform four dimensional assessments on 50 examples sampled from the KP20k test set. To ensure high inter annotator agreement, we drafted a standardized evaluation guideline and provided prototypical scoring examples; Tab. 5 shows the manual evaluation instructions for the coverage dimension. Since the human scores range over the ordered integers 1–5, we compute inter annotator reliability using Krippendorff's alpha. The resulting alpha values are 0.85 for coverage, 0.74 for distinctiveness, 0.62 for conciseness, and 0.79 for diversity, indicating a high level of agreement among the evaluators.

6.6 Consistency between CAME and Human Evaluation

In this section, we conduct a series of experimental analyses on the four dimensional CAME framework—Coverage (COV), Distinctiveness (DIS), Conciseness (CON), and Diversity (DIV)—including (1) consistency between CAME and expert judgments, (2) inter model consistency among CAME implementations using different large language models, and (3) CAME's measurement capability for keyphrase generation models.

We first evaluate the 50 zero shot keyphrase sets generated by GPT 3.5 turbo on the KP20k test set using three baseline metrics, CAME-Base, and five LLM based CAME variants. We then compute the consistency between these automatic scores and expert ratings, as shown in Tab. 6 (with boldface indicating the best score per column and underlining the second best). We measure consistency using Spearman's rank correlation coefficient (ρ) and Kendall's Tau (τ). Spearman's ρ assesses the strength of a monotonic relationship between two variables, making it appropriate for ordinal data or distributions that deviate from normality. Kendall's τ evaluates the association between two ranked variables and is particularly robust for small samples or when many tied ranks occur. Both coefficients range from –1 to 1, where 1 signifies perfect positive monotonic agreement and –1 perfect negative agreement.

The experimental results demonstrate that CAME Base, which relies on traditional computational methods, consistently outperforms the baseline metrics across all four dimensions. Moreover, the LLM based CAME variants achieve even higher agreement with expert judgments than CAME Base. In particular, CAME GPT 4o and CAME DeepSeek V3 exhibit the strongest concordance: CAME DeepSeek V3 attains a Spearman's ρ of 0.78 on the coverage dimension. Because coverage and diversity are relatively straightforward to assess, both CAME Base and the various LLM based CAME implementations show

Table 5: Guidelines for manual evaluation of coverage

Task Description: You will be given a passage of text along with a set of keyphrases generated by a model. Your job is to assess how well the keyphrase set adheres to the given evaluation metric. The scoring guidelines are:
1 point: The keyphrase set *hardly follows the evaluation metric.*
2 points: The keyphrase set *follows the evaluation metric to a low degree.*
3 points: The keyphrase set *roughly follows the evaluation metric.*
4 points: The keyphrase set *largely follows the evaluation metric.*
5 points: The keyphrase set *fully follows the evaluation metric.*
Evaluation Metric:
Coverage The keyphrase set should cover all important information discussed in the document. If it omits major aspects of the document, it should be penalized; if it includes all aspects, it should be rewarded.
Example:
Text: Word Sense Disambiguation (WSD) for event trigger detection. This paper introduces a WSD-based method for detecting event triggers in biomedical text. We first examine the applicability of existing WSD techniques to trigger disambiguation on the BioNLP 2009 shared task dataset, and find that on certain word classes we can outperform traditional CRF-based approaches. Based on this finding, we combine WSD with CRF and achieve significant improvements over standalone CRF, especially in recall.
Keyphrases: biomedical text; machine learning; information extraction
Score: 2
Rationale: The keyphrase set captures the application domain, but severely omits methodological details—it fails to include WSD, CRF, or their combined approach, and thus does not reflect the paper's core contributions.

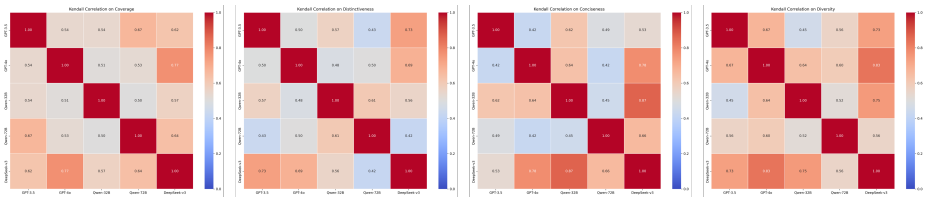
markedly improved alignment with human ratings on these dimensions. By contrast, evaluating distinctiveness and conciseness—which require deeper domain expertise—yields lower agreement overall; only the GPT 4o and DeepSeek V3 backbones achieve comparatively strong correlations here. Among the baseline metrics, SemF1—which measures semantic overlap between generated and reference keyphrase sets—performs better on coverage than the other baselines, but still falls short of CAME Base. We attribute this to SemF1's sensitivity to the specific reference set, whereas CAME Base evaluates coverage directly against the source context and so offers a more objective measure. Given the computational complexity and higher performance of the LLM based variants, we focus our subsequent analyses on CAME implementations built atop large language models.

Table 6: Spearman (ρ) and Kendall Tau (τ) Correlation Coefficients Between Automatic and Manual Evaluation Metrics.

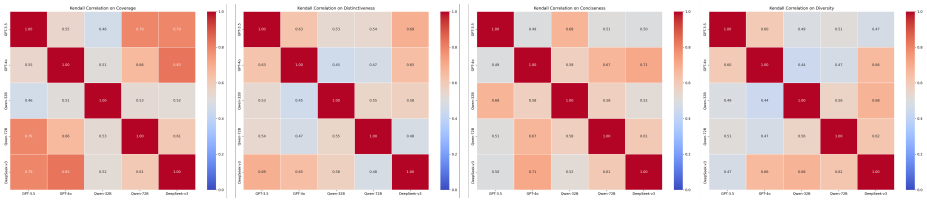
	COV (↑)		**DIS** (↑)		**CON** (↑)		**DIV** (↑)	
Metric	ρ	τ	ρ	τ	ρ	τ	ρ	τ
F1@5	0.15	0.13	0.11	0.09	0.12	0.08	0.15	0.11
F1@10	0.23	0.17	0.10	0.07	0.15	0.11	0.18	0.14
SemF1	0.33	0.30	0.17	0.12	0.20	0.17	0.22	0.19
CAME-Base	0.41	0.36	0.32	0.29	0.34	0.31	0.44	0.37
CAME-GPT-3.5-turbo	0.67	0.59	0.55	0.47	0.58	0.52	0.63	0.57
CAME-GPT-4o	0.72	**0.64**	**0.69**	**0.55**	**0.65**	**0.61**	0.70	0.62
CAME-Qwen2.5-32B	0.55	0.44	0.50	0.42	0.47	0.38	0.56	0.49
CAME-Qwen2.5-72B	0.64	0.57	0.52	0.44	0.50	0.44	0.71	0.66
CAME-DeepSeek-V3	**0.78**	**0.64**	0.67	**0.55**	0.63	0.56	**0.75**	**0.68**

6.7 Consistency of CAME Across Diverse LLMs

CAME's cross-model consistency is evaluated by measuring pairwise Kendall's τ correlations among the scores generated by GPT-3.5-turbo, GPT-4o, Qwen2.5-32B-Instruct, Qwen2.5-72B-Instruct, and DeepSeek-V3. This analysis evaluates the stability of the CAME framework when different LLMs serve as its backbone. The resulting heat maps are presented in Fig. 7.



(a) Consistency Analysis on MultipartiteRank results



(b) Consistency Analysis on GPT-3.5-turbo(0-shot) results

Fig. 7: Heat map of the Kendall Tau (τ)of evaluation between CAME implemented on different LLM

Fig. 7a shows the Kendall Tau coefficients for evaluations across four dimensions of extraction results from the MultipartiteRank model using different large language models; the second row displays Kendall Tau coefficients for evaluations across four dimensions of zero-shot generation results from the GPT-3.5-turbo model using different large language models. As revealed by the heatmap, evaluations of Coverage and Diversity achieve relatively high consistency across different language models, while cross-LLM consistency for Uniqueness and Conciseness is weaker. CAME implementations based on GPT-4o and DeepSeek-V3 demonstrate comparatively higher evaluation consistency, likely because language models with larger parameter counts embed richer linguistic knowledge, enabling more precise determination of phrase domain relevance and structural attributes, thus yielding more stable and accurate evaluations. Overall, CAME exhibits strong generalization across different large language models.

6.8 Discussion of LLM-based Evaluation

While LLMs provide a more comprehensive assessment of keyphrase prediction by capturing semantic and contextual relevance, their evaluation results can vary across different models due to inherent model biases. In contrast, traditional metrics in the surveys [70] such as precision, recall, and F1 are based on well-defined, reproducible formulas that ensure consistency and objectivity.

Furthermore, LLMs as black-box models, where the reasoning behind their evaluations is opaque. This lack of transparency makes it difficult to interpret the evaluations, unlike traditional metrics, which provide clear and actionable feedback based on established rules.

For these reasons, we treat LLMs based scores as complementary signals rather than replacements for statistically grounded metrics. We propose further work to explore a systematic combination of LLMs evaluation with human annotation and traditional metrics.

7 Challenges and Future Directions

7.1 Specialty Keyphrase Prediction

Challenge. Keyphrase generation in specialized domains poses additional challenges due to highly domain-specific terminology, jargon, and conceptual knowledge. General-purpose LLMs are typically pretrained on broad, general-domain corpora, resulting in limited exposure to domain-specific vocabulary, abbreviations, and complex conceptual relationships. The difficulty is further exacerbated by the scarcity of annotated datasets, frequent updates to terminology, and the dependence of many domain-specific keyphrases on background knowledge, such as clinical guidelines or legal precedents. Moreover, models trained in a specific subdomain often struggle to generalize across other subdomains, highlighting the issue of domain shift.

Future directions. In scenarios with sufficient computational resources, using available labeled datasets or constructing pseudo-labeled data to fine-tune domain-specific keyphrase prediction models. Constructing high-quality pseudo-labels and applying efficient fine-tuning strategies can prevent model degradation and enhance adaptation efficiency, representing a promising research direction.

In resource-constrained scenarios, introducing domain knowledge as adapters to improve keyphrase extraction performance becomes essential. Future research should focus on effective organization of external knowledge and seamless integration of domain-specific knowledge into LLMs to maximize performance gains without relying heavily on large-scale fine-tuning.

7.2 Trustworthiness and Hallucination Control

Challenge. Despite the strong potential of LLMs in generating absent keyphrases, models are prone to hallucination, producing keyphrases that are irrelevant or factually incorrect, which undermines the reliability of outputs. This issue is especially critical in scientific, medical, or legal applications, where the trustworthiness of predictions is paramount.

Future directions. Promising directions to address these limitations involve multiple complementary strategies. First, more effective prompt engineering techniques can be employed to better guide the LLMs in generating conceptually novel and contextually relevant absent keyphrases, reducing superficial paraphrasing. Second, integrating retrieval-augmented methods allows models to ground their outputs in external knowledge sources, such as domain-specific corpora or knowledge graphs, thereby enhancing semantic coverage and reducing hallucinations. Third, robust post-generation verification mechanisms can be introduced to filter, correct, or re-rank low-quality predictions. These mechanisms may leverage uncertainty estimation, semantic consistency checks, or alignment with external knowledge to identify and mitigate unreliable outputs. Additionally, multi-dimensional evaluation metrics—including coverage, distinctiveness, and conciseness—can inform adaptive post-processing, further improving the quality of generated keyphrases. Collectively, these strategies aim to increase the reliability, informativeness, and practical applicability of keyphrase prediction in real-world scenarios.

7.3 Evaluation-Guided Keyphrase Generation

Challenge. Current unsupervised and LLM-based keyphrase generation systems predominantly operate in a fully automated manner, often decoupling evaluation from generation. As a result, feedback from evaluation—whether automatic metrics or human assessment—is rarely utilized to refine or guide the generation process. This disconnect limits the adaptability and user alignment of generated keyphrases, which are intended for human-centric tasks such as retrieval, summarization, and indexing.

Future directions. A promising direction for advancing keyphrase prediction is to integrate evaluation feedback directly into the generation process. Instead of treating evaluation as a post hoc step, models can iteratively leverage evaluation signals to refine outputs, ensuring better alignment with linguistic properties and user expectations. Adaptive frameworks can incorporate evaluation metrics as optimization objectives, enabling dynamic adjustment during generation. Such evaluation-guided paradigms hold potential to bridge the gap between automated prediction and human judgment, ultimately fostering more reliable, adaptive, and user-aligned keyphrase generation systems.

8 Conclusion

This survey provides a comprehensive overview of recent advances in unsupervised keyphrase prediction and evaluation, highlighting the challenges and opportunities in the era of large language models. We analyze the linguistic properties of keyphrases to inform model design and evaluation strategies, and review methods spanning statistical approaches to LLMs, with particular attention to emerging techniques. Furthermore, we systematically examine existing evaluation metrics, identify their limitations, and introduce a novel reference-free metric, CAME, whose effectiveness is validated through experiments. Finally, we outline promising research directions, aiming to facilitate the development of more robust, adaptable, and human-aligned keyphrase prediction systems.

Acknowledgements This work was supported by the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007), the National Natural Science Foundation of China (62376138).

References

1. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. arXiv preprint arXiv:1704.02853 (2017)
2. Bai, X., Wu, X., Stojkovic, I., Tsioutsiouliklis, K.: Leveraging large language models for improving keyphrase generation for contextual targeting. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. pp. 4349–4357 (2024)
3. Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M., et al.: Unilmv2: Pseudo-masked language models for unified language model pre-training. In: International conference on machine learning. pp. 642–652. PMLR (2020)
4. Belz, A., Thomson, C., Reiter, E., Mille, S.: Non-repeatable experiments and non-reproducible results: The reproducibility crisis in human evaluation in nlp. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 3676–3687 (2023)
5. Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., Jaggi, M.: Simple unsupervised keyphrase extraction using sentence embeddings. arXiv preprint arXiv:1801.04470 (2018)

6. Boudin, F.: Pke: an open source python-based keyphrase extraction toolkit. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations. pp. 69–73 (2016)
7. Boudin, F.: Unsupervised keyphrase extraction with multipartite graphs. arXiv preprint arXiv:1803.08721 (2018)
8. Boudin, F., Aizawa, A.: Unsupervised domain adaptation for keyphrase generation using citation contexts. arXiv preprint arXiv:2409.13266 (2024)
9. Boudin, F., Gallina, Y.: Redefining absent keyphrases and their effect on retrieval effectiveness. arXiv preprint arXiv:2103.12440 (2021)
10. Boudin, F., Gallina, Y., Aizawa, A.: Keyphrase generation for scientific document retrieval. arXiv preprint arXiv:2106.14726 (2021)
11. Bougouin, A., Boudin, F., Daille, B.: Topicrank: Graph-based topic ranking for keyphrase extraction. In: International Joint Conference on Natural Language Processing (2013)
12. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A.: Yake! keyword extraction from single documents using multiple local features. Information Sciences **509**, 257–289 (2020)
13. Chen, Y., Wang, R., Jiang, H., Shi, S., Xu, R.: Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In: IJCNLP (Findings) (2023)
14. Chiang, C.H., Lee, H.y.: A closer look into using large language models for automatic evaluation. In: Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 8928–8942 (2023)
15. Church, K., Hanks, P.: Word association norms, mutual information, and lexicography. Computational linguistics **16**(1), 22–29 (1990)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv **abs/1810.04805** (2019)
17. Ding, H., Luo, X.: Attentionrank: Unsupervised keyphrase extraction using self and cross attentions. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 1919–1928 (2021)
18. Do, L.T., Akash, P.S., Chang, K.C.C.: Unsupervised open-domain keyphrase generation. arXiv preprint arXiv:2306.10755 (2023)
19. Firoozeh, N., Nazarenko, A., Alizon, F., Daille, B.: Keyword extraction: Issues and methods. Natural Language Engineering **26**(3), 259–291 (2020)
20. Florescu, C., Caragea, C.: 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). pp. 1105–1115 (2017)
21. Gallina, Y., Boudin, F., Daille, B.: Kptimes: A large-scale dataset for keyphrase generation on news documents. arXiv preprint arXiv:1911.12559 (2019)
22. Gollapalli, S.D., Caragea, C.: Extracting keyphrases from research papers using citation networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 28 (2014)
23. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
24. Houbre, M., Boudin, F., Daille, B.: A large-scale dataset for biomedical keyphrase generation. arXiv preprint arXiv:2211.12124 (2022)
25. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Conference on Empirical Methods in Natural Language Processing (2003)

26. Joshi, R., Balachandran, V., Saldanha, E., Glenski, M., Volkova, S., Tsvetkov, Y.: Unsupervised keyphrase extraction via interpretable neural networks. arXiv preprint arXiv:2203.07640 (2022)
27. Kang, B., Shin, Y.: Samrank: Unsupervised keyphrase extraction using self-attention map in bert and gpt-2. In: The 2023 Conference on Empirical Methods in Natural Language Processing (2023)
28. Kang, B., Shin, Y.: Improving low-resource keyphrase generation through unsupervised title phrase generation. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). pp. 8853–8865 (2024)
29. Kang, B., Shin, Y.: Empirical study of zero-shot keyphrase extraction with large language models. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 3670–3686 (2025)
30. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In: *SEMEVAL (2010)
31. Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., Bai, X.: Promptrank: Unsupervised keyphrase extraction using prompt. arXiv preprint arXiv:2305.04490 (2023)
32. Krapivin, M., Autaeu, A., Marchese, M., et al.: Large dataset for keyphrases extraction (2009)
33. Kulkarni, M., Mahata, D., Arora, R., Bhowmik, R.: Learning rich representation of keyphrases from text. arXiv preprint arXiv:2112.08547 (2021)
34. Kundu, T., Chowdhury, J.R., Caragea, C.: Neural keyphrase generation: Analysis and evaluation. arXiv preprint arXiv:2304.13883 (2023)
35. Liang, X., Wu, S., Li, M., Li, Z.: Unsupervised keyphrase extraction by jointly modeling local and global context. arXiv preprint arXiv:2109.07293 (2021)
36. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-eval: Nlg evaluation using gpt-4 with better human alignment. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 2511–2522 (2023)
37. Mahata, D., Agarwal, N., Gautam, D., Kumar, A., Parekh, S., Singla, Y.K., Acharya, A., Shah, R.R.: Ldkp: A dataset for identifying keyphrases from long scientific documents. arXiv preprint arXiv:2203.15349 (2022)
38. Maragheh, R.Y., Fang, C., Irugu, C.C., Parikh, P., Cho, J., Xu, J., Sukumar, S., Patel, M., Korpeoglu, E., Kumar, S., et al.: Llm-take: Theme-aware keyword extraction using large language models. In: 2023 IEEE International Conference on Big Data (BigData). pp. 4318–4324. IEEE (2023)
39. Martínez-Cruz, R., López-López, A.J., Portela, J.: Chatgpt vs state-of-the-art models: a benchmarking study in keyphrase generation task. arXiv preprint arXiv:2304.14177 (2023)
40. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. arXiv preprint arXiv:1704.06879 (2017)
41. Merrouni, Z.A., Frikh, B., Ouhbi, B.: Hake: an unsupervised approach to automatic keyphrase extraction for multiple domains. Cognitive Computation **14**(2), 852–874 (2022)
42. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. pp. 404–411 (2004)
43. Mohan, J., Chowdhury, J.R., Caragea, T.M.C.: Zero-shot keyphrase generation: Investigating specialized instructions and multi-sample aggregation on large language models

44. Nguyen, T.D., Kan, M.Y.: Keyphrase extraction in scientific publications. In: International conference on Asian digital libraries. pp. 317–326. Springer (2007)
45. Nguyen, T.D., Kan, M.Y.: Keyphrase extraction in scientific publications. In: International Conference on Asian Digital Libraries (2007)
46. Piedboeuf, F., Langlais, P.: A new dataset for multilingual keyphrase generation. Advances in Neural Information Processing Systems **35**, 38046–38059 (2022)
47. Powers, D.: Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies **2**(1), 37–63 (2011)
48. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
49. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
50. Salaün, O., Piedboeuf, F., Berre, G.L., Hermelo, D.A., Langlais, P.: Europa: A legal multilingual keyphrase generation dataset. arXiv preprint arXiv:2403.00252 (2024)
51. Shao, L., Zhang, L., Peng, M., Ma, G., Yue, H., Sun, M., Su, J.: One2set+ large language model: Best partners for keyphrase generation. arXiv preprint arXiv:2410.03421 (2024)
52. Shen, L., Liu, L., Jiang, H., Shi, S.: On the evaluation metrics for paraphrase generation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 3178–3190 (2022)
53. Shen, X., Wang, Y., Meng, R., Shang, J.: Unsupervised deep keyphrase generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 11303–11311 (2022)
54. Song, M., Feng, Y., Jing, L.: A preliminary empirical study on prompt-based unsupervised keyphrase extraction (2024), `https://arxiv.org/abs/2405.16571`
55. Song, M., Geng, X., Yao, S., Lu, S., Feng, Y., Jing, L.: Large language models as zero-shot keyphrase extractor: A preliminary empirical study. arXiv preprint arXiv:2312.15156 (2023)
56. Song, M., Jiang, H., Shi, S., Yao, S., Lu, S., Feng, Y., Liu, H., Jing, L.: Is chatgpt a good keyphrase generator? a preliminary study. arXiv preprint arXiv:2303.13001 (2023)
57. Song, M., Liu, H., Feng, Y., Jing, L.: Improving embedding-based unsupervised keyphrase extraction by incorporating structural information. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 1041–1048 (2023)
58. Song, M., Liu, H., Jing, L.: Hyperrank: hyperbolic ranking model for unsupervised keyphrase extraction. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 16070–16080 (2023)
59. Song, M., Xu, P., Feng, Y., Liu, H., Jing, L.: Mitigating over-generation for unsupervised keyphrase extraction with heterogeneous centrality detection. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 16349–16359 (2023)
60. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of documentation **28**(1), 11–21 (1972)
61. Sun, Y., Qiu, H., Zheng, Y., Wang, Z., Zhang, C.: Sifrank: a new baseline for unsupervised keyphrase extraction based on pre-trained language model. IEEE Access **8**, 10896–10906 (2020)
62. Venturinia, S., Kinkelb, S.: Meaning in order, order in meaning: Semantic r-precision for keyphrase evaluation

63. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: AAAI Conference on Artificial Intelligence (2008)
64. Wang, J., Liang, Y., Meng, F., Sun, Z., Shi, H., Li, Z., Xu, J., Qu, J., Zhou, J.: Is chatgpt a good nlg evaluator? a preliminary study. arXiv preprint arXiv:2303.04048 (2023)
65. Wang, S., Dai, S., Jiang, J.: Thinking like an author: A zero-shot learning approach to keyphrase generation with large language model. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 335–350. Springer (2024)
66. Wang, Y., Li, J., Lyu, M.R., King, I.: Cross-media keyphrase prediction: A unified framework with multi-modality multi-head attention and image wordings. arXiv preprint arXiv:2011.01565 (2020)
67. Wu, D., Ahmad, W.U., Chang, K.W.: Rethinking model selection and decoding for keyphrase generation with pre-trained sequence-to-sequence models. arXiv preprint arXiv:2310.06374 (2023)
68. Wu, D., Shen, X., Chang, K.W.: Metakp: On-demand keyphrase generation. arXiv preprint arXiv:2407.00191 (2024)
69. Wu, D., Yin, D., Chang, K.W.: Kpeval: Towards fine-grained semantic-based keyphrase evaluation. arXiv preprint arXiv:2303.15422 (2023)
70. Xie, B., Song, J., Shao, L., Wu, S., Wei, X., Yang, B., Lin, H., Xie, J., Su, J.: From statistical methods to deep learning, automatic keyphrase prediction: A survey. Information Processing & Management **60**(4), 103382 (2023)
71. Xiong, L., Hu, C., Xiong, C., Campos, D., Overwijk, A.: Open domain web keyphrase extraction beyond language modeling. arXiv preprint arXiv:1911.02671 (2019)
72. Yuan, X., Wang, T., Meng, R., Thaker, K., Brusilovsky, P., He, D., Trischler, A.: One size does not fit all: Generating and evaluating variable number of keyphrases. arXiv preprint arXiv:1810.05241 (2018)
73. Zanutto, D.: Leveraging llm-generated keyphrases and clustering techniques for topic identification in product reviews (2023)
74. Zhang, L., Chen, Q., Wang, W., Deng, C., Zhang, S., Li, B., Wang, W., Cao, X.: Mderank: A masked document embedding rank approach for unsupervised keyphrase extraction. arXiv preprint arXiv:2110.06651 (2021)
75. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
76. Zhang, Z., Liang, X., Zuo, Y., Lin, C.: Improving unsupervised keyphrase extraction by modeling hierarchical multi-granularity features. Information Processing & Management **60**(4), 103356 (2023)