FEM-KP: A Functional Evaluation Metric for Keyphrase Prediction Models

Lahbib Ajallouda[1], Ahmed Zellou[2]
LARSLACH Laboratory Poly-Disciplinary Faculty of Es-Semara, Ibn Zohr University, Agadir, Morocco[1]
SPM Research Team ENSIAS, Mohammed V University, Rabat, Morocco[2]

***Abstract*—Keyphrase prediction models are among the natural language processing (NLP) tasks that have improved their performance with transformers and large language models (LLMs). Instead of extracting present keyphrases in the text, these models also generate absent keyphrases. This improvement has led to significant challenges in the evaluation process of these models, which rely on metrics that compare the predicted keyphrases with the reference keyphrases. To measure the performance of these models, several evaluation metrics such as F1-score, ROUGE-L, and BertScore have been used. However, they often prioritize lexical similarity over semantic usefulness. Consequently, the functional usefulness of keyphrases in document representation is not evaluated during the evaluation process, which leads to inconsistencies in the evaluation results. Therefore, in this paper we propose a functional evaluation metric for Keyphrase prediction models (FEM-KP), a new evaluation metric that uses a two-track approach where track (A) evaluates the performance of the model to generate keyphrases capable of constructing a document summary, while track (B) measures the ability of these phrases to retrieve the document. We evaluated the performance of four keyphrase prediction models using current evaluation metrics and FEM-KP across the Inspec, KP20k, and Krapivin datasets. The experimental results showed that FEM-KP is the only evaluation system that maintained a consistent performance ranking regardless of document length or dataset complexity. In contrast, other metrics showed inversions in ranking. These results confirm that FEM-KP is a robust, reliable, and domain-independent evaluation metric for evaluating the performance of keyphrase prediction systems.**

Keywords*—Document retrieval; Document summarization; Evaluation metrics; Functional evaluation metric; Keyphrase prediction models; natural language processing

I. Introduction

Keyphrases prediction is one of the most important tasks in natural language processing [1]. These phrases provide descriptive and semantic information that summarizes the content of documents. Recent studies have demonstrated that exploiting keyphrases improves the performance of summarization models by directing them toward the important topics of the text [2]. Also, the exploitation of keyphrases in the retrieval task improves the quality of retrieved snippets and reduces computational costs [3]. Keyphrases also improve the accuracy of responses in question-and-answer tasks [4]. Furthermore, digital indexation has been improved by using keyphrase generation [5].

Keyphrase prediction models (KPM) are typically evaluated by comparing predicted with referenced keyphrases, without considering semantic aspects [6]. This evaluation does not accurately reflect the performance of these models (KPEVAL). Generative models are more affected by this problem because they generate the keyphrases rather than extracting them from the text [7].

To improve the quality of KPM evaluation, some studies, such as [8], have proposed a framework exploit edit distance, token matching, and duplication penalty. Also, at [6], a two-way comparison was conducted between predicted and reference keyphrases using the BertScore [9]. In contrast, the KPEVAL framework propose to combine accuracy, variety, reliability, and reference consistency. However, several challenges still limit the reliability of the results of these models, such as the reliance on a set of reference keyphrases to compare KPM performance that are often self-defined and influenced by linguistic style. Also, the difficulty to evaluate the functional performance of the keyphrases generated in the document representation.

To overcome these challenges, we propose the FEM-KP metric, a comprehensive metric to evaluate KPM by measuring the functional utility of these phrases rather than linguistic or semantic similarity. The importance of FEM-KP lies in transforming the evaluation process, which focuses on linguistic or semantic similarity between the predicted and reference keyphrases, into a deeper semantic evaluation of the functional quality of these phrases. Our metric adopts two tracks, the first evaluates the ability of the predected keyphrases to generate a document summary, while the second evaluates their ability to retrieve the document.

The content of paper includes, a review of previous studies that focused on keyphrase prediction models, text summarization, text retrieval, and sentence embedding techniques in section 2. The current evaluation methods for keyphrase prediction models are presented in section 3, including evaluation metrics, evaluation datasets, and challenges that limit their reliability. Section 4 provides the proposed FEM-KP metric, its design, and formulation. The experimental results of FEM-KP performance are presented in Section 5. The paper concludes with section 6, which summarizes our work and identifies future research directions.

II. Related Work

To present the mechanisms and techniques that will be exploited in FEM-KP metric, we have devoted this section to review the keyphrase prediction models, sentence embedding techniques, text retrieval techniques, and text summarization techniques.

A. Keyphrase Prediction Models

Keyphrases are expressions consisting of one or more words, that serve not only to index but also as semantic summaries. Keyphrases are exploited in classification, clustering, text summarization, and information retrieval [10]. Keyphrase models have evolved from extractive models based on statistical techniques to generative models based on transformers and large language models (LLMs) [11]. Several reviews have addressed the development of these models, while the review in [12] is focused on statistical and graphical methods, [13] reviewed the improved performance using deep learning to generate keyphrases.

In contrast, other studies, such as [14], focused on reviewing the latest methods that use pre-trained models such as SBERT and RoberTa for keyphrase extraction, and how these models improved extraction results compared to previous methods. In [15], the comparison between extraction and generation mechanism, and the advantages and disadvantages of statistical methods and deep learning methods to extract and generate keyphrases were identified. Furthermore, [16] demonstrated how sentence embeddings can be used to improve contextual extraction performance. Also, [17] reviewed the integration of extraction and generation mechanisms in keyword prediction, proposing a classification of models based on pre-trained models, as well as semi-supervised and unsupervised approaches. In contrast, [18] analyzes the performance of LLM models to extract keyphrases and presents a comparison between Llama3, GPT-4o and other large language models.

B. Text Semantic Representation

With their crucial role in many NLP tasks, text embedding models have garnered increasing attention from researchers, particularly with the emergence of Large Language Models (LLMs) [19]. Text embedding began with models that represented each word individually, such as Word2Vec, which learned word representations from their contexts. Later, GloVe (Global Vectors) exploit also comprehensive word frequency statistics to improve the performance of embedding. However, the challenge of representing words according to their meaning within a sentence remained until the emergence of ELMo as a model based on bidirectional LSTM networks to produce contextual representations based on word meaning within a sentence.

The advent of transformers revolutionized text embedding, with the emergence of the BERT (Bidirectional Coder Representations of Transformers) model, which uses transformer architecture to create deep, and contextual embeddings of words and sentences. This model was further enhanced at the sentence and paragraph by the SBERT model. Multilingual models like LaBSE, which produces comparable embeddings across different languages, have also been developed, proving very useful for multilingual retrieval. Furthermore, the challenge of long documents has led to the emergence of models that represent long documents, such as Dense Passage Retrieval (DPR), which uses BERT-like models to embed long documents. ColBERTv2 also produces efficient embeds by layering the representation to enhance semantic differentiation. In contrast, the emergence of Large Language Models (LLMs) such as GPT4 and LLaMA has contributed to the production of text embeds that exhibit a greater ability to capture semantic relationships, both in retrieval and text generation. OpenAI Embeddings relies on LLM models to generate vectors used in semantic retrieval, Q&A, and classification.

The open-source LLaMA Embeddings is also used to create embeds with greater context and a high capacity for representing long texts. Fig. 1 presents the evolution of text embedding models.

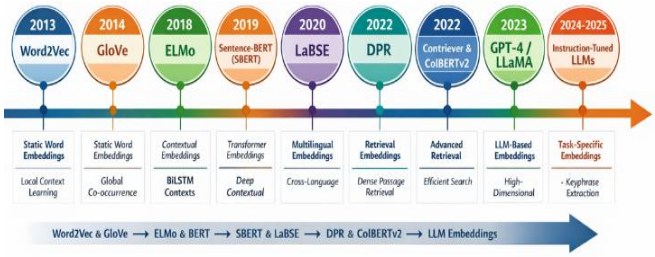

Fig. 1. Evolution of text embedding models.

C. Text Summarization Models

Text summarization tasks have evolved through several approaches. Extractive summarization, which relies on features such as word frequency, position, and similarity metrics, was the first method used text summarization [20]. Graph algorithms such as TextRank and LexRank [21] were also exploited to model the relationships between sentences. With the advent of neural models, particularly RNN/LSTM, abstractive summarization emerged, which focused on paraphrasing the text rather than extracting [22], [23]. The ability of transformers, such as BERT [24] and BART [25], to provide deep contextual understanding and generate texts has also contributed to improving the text summarization task. Large language models (LLMs) such as GPT-3 and GPT-4 have also enhanced text summarization performance [26]. Several reviews have focused on the text summarization task. The impact of large language models on summary coherence in zero-based training environments was studied in [27]. The use of LLMs in the specialize contexts, such as the medical or scientific fields, was also discussed in the studies [28] and [29]. These studies confirmed the significant impact of LLMs on text summarization task.

D. Text Retrieval Models

Text retrieval models are classified into several categories. The first is lexical and statistical matching, which relies on term-frequency-inverse document frequency (TF-IDF) to sort documents, such as BM25 [30], which sorts text based on keyphrases. There is also the category of dense retrieval, which uses transforms to represent text in a vector space, such as the

DPR (Dense Passage Retrieval) model [31] , which exploits bi-encoder to represent the query and the document separately. Also, a hybrid and multi-vector late interaction models, which combine the efficiency of dense retrieval models with the accuracy of lexical retrieval, such as the ColBERT (Contextualized Late Interaction over BERT) model [32], which is more accurate in capturing fine-grained textual details.

Large language models have also contributed to improved retrieval task. Large Language Model-Augmented Retrieval (LLM-Augmented Retrieval), which focuses on using LLMs to generate hypothetical documents or optimize queries before the retrieval process, such as HyDE (Hypothetical Document Embeddings) model [33]. This model generates a hypothetical text from the query and then searches for similar documents. In general, studies indicate that retrieval applications rely on pre-trained embeddings and neural structures to provide more accurate semantic matching [34] and [35], which supports the hypothesis that a keyphrase is a condensed summary from which the original document can be retrieved and reconstructed.

Word embedding models (Word2Vec, GloVe) are considered the first step towards text embedding, followed by the emergence of contextual models (ELMo, BERT), which evolved into sentence and paragraph embedding models (SBERT, LaBSE), then expanded further to contextual retrieval models (DPR, ColBERTv2), and finally to LLM-based embeddings, which are related to the higher text representation capabilities of multiple NLP tasks.

III. Keyphrase Evaluation Paradigms

In recent years, most keyphrase prediction models have adopted a generation mechanism, which has complicated the task of evaluating these models. Most current evaluation metrics rely on lexical matching, which lacks the ability to measure semantic meaning. In this context, this section will provide a review of current evaluation metrics, from statistical to embedding metrics, to highlight the research gap represented by the absence of a standardized metric that combines the precision with the contextual and functional significance of predicted keyphrase.

A. Lexical Evaluation Metrics

Lexical metrics are the most commonly used to evaluate keyphrase prediction models. These metrics calculate the exact match between the keyphrases predicted by the model (KP) and the reference keyphrases (RP).

1) Precision, recall, and f1-measure metrics: Precision, Recall, and F1-measure are used to evaluate keyphrase prediction models based on exact matching. Precision measures the accuracy of predictions, by calculating the proportion of correct phrases extracted from the total identified by the model. It also indicates the model's ability to exclude irrelevant phrases. Recall measures the coverage of reference phrases, by calculating the model's ability to retrieve all reference keyphrases. F1-measure measures the harmonic mean between precision and recall, to achieve a balance between the coverage and the accuracy. Table I presents the mathematical definitions of Precision, Recall, and F1-measure metrics.

TABLE I. Mathematical Formula of Precision, Recall, and F1-Measure Metrics

Metric	Mathematical Formula
Precision	$\frac{\text{correctly predicted keyphrases}}{\text{all predicted keyphrases}}$ (1)
Recall	$\frac{\text{correctly predicted keyphrases}}{\text{all reference keyphrases}}$ (2)
F1-measure	$2 \times \frac{Precision \times Recall}{Precision + Recall}$ (3)

Although the precision, recall, and F1-Measure metrics provide a lexical evaluation of key phrase prediction performance, they rely on exact matching between the predicted and reference keyphrases. However, lexical interference at the n-gram and sequences, which results in partial matching between phrases, is the most significant challenge for these metrics.

2) ROUGE metric: The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric, primarily used to evaluate text summarization models. This metric has been adapted to evaluate keyphrase prediction by treating each predicted keyphrase and reference keyphrase as a sequence of tokens. ROUGE calculates a lexical overlap between predicted keyphrases and reference keyphrases. REUGE is used in various forms. The most commonly used variants are:

ROUGE-N: measures the overlap of n-grams. For example, ROUGE-1 evaluates unigram overlap, and ROUGE-2 evaluates bigram overlap. ROUGE-N is calculated using formula 4.

$$ROUGE - N = \frac{\sum_{gram_n \in Ref} min\,(Count_{pred}(gram_n), Count_{ref}(gram_n))}{\sum_{gram_n \in Ref} Count_{ref}(gram_n))} \quad (4)$$

where:

- Ref: Set of reference keyphrases.
- Pred: set of predicted keyphrases.
- $gram_n$: n-gram token sequence.
- $Count_{pred}(.)$: Occurrences of the n-gram in predicted keyphrases.
- $Count_{ref}(.)$: Occurrences of the n-gram in reference keyphrases.

ROUGE-L: Unlike ROUGE-N, ROUGE-L relies to find the Longest Common Subsequence (LCS) that appear in both the predicted keyphrase and the reference keyphrase. ROUGE-L is calculated using LCS-based decision and recall metrics by formulas 5, 6 and 7.

$$Precision_{LCS} = \frac{LCS(PK,RK)}{|PK|} \quad (5)$$

$$Recall_{LCS} = \frac{LCS(PK,RK)}{|RK|} \quad (6)$$

$$ROUGE - L_{LCS} = \frac{(1+\beta^2) \times Precision_{LCS} \times Recall_{LCS}}{Recall_{LCS} + \beta^2 \times Precision_{LCS}} \quad (7)$$

where:

- RK : reference keyphrases
- PK : predicted keyphrases
- LCS(PK, RK): length of the longest common subsequence between PK and RK.
- |RK| : total number of tokens in reference keyphrases
- |PK| : total number of tokens in predicted keyphrases
- β :A parameter used to set the priority between precision and recall (typically β=1).

While lexical metrics facilitate the evaluation process, they disregard the expected order and importance of keyphrases in the document. This limitation has spurred the use rank-based metrics such as MRR, NDCG, and MAP, which consider the position and importance of the keyphrase.

B. Rank-Based Evaluation Metrics

To overcome the challenges of lexical evaluation, rank-based metrics consider the rinking of predicted keyphrases, where top-ranked candidates are of primary interest. Mean Reciprocal Rank (MRR) is a simple metric that focuses on the rank of the first keyphrase predicted $rank_i$ across a set of N documents. MRR is calculated by formula 8.

$$MRR = \frac{1}{|N|}\sum_{i=1}^{|N|}\frac{1}{rank_i} \quad (8)$$

MRR ignores the rank of other keyphrases is a challenge that limits its ability to evaluate multiple keyphrases. To address this challenge, Mean Average Precision (MAP) metric calculates the average precision for each position where keyphrase occurs. Formula 9 is used to calculate the MAP metric.

$$MAP = \frac{1}{|N|}\sum_{i=1}^{|N|}\left(\frac{1}{|K_i|}\sum_{j=1}^{m}P_i(j)\times rel_i(j)\right) \quad (9)$$

where K_i is the set of reference keyphrases, m is the number of predicted key phrases, $P_i(j)$ denotes precision at rank k, and $rel_i(j)$ is a relevance indicator, is 1 if the predicted keyphrase at rank k is relevant, and 0 otherwise.

MAP metric treats all keyphrases with the same weight, regardless of their importance in the document. To overcome this challenge, the Normalized Discounted Cumulative Gain (NDCG) metric, uses the logarithmic discount, where the importance of the keyphrase decreases as its position in the list decreases. Formula 10 is used to calculate the NDCG metric.

$$\mathrm{NDCG_k} = \frac{\sum_{i=1}^{k}\frac{2^{\mathrm{rel_{i-1}}}}{\log_2(i+1)}}{\sum_{i=1}^{k}\frac{2^{\mathrm{Irel_{i-1}}}}{\log_2(i+1)}} \quad (10)$$

where:

- k: The number of top predicted keyphrases.
- i: The rank position in the predicted list.
- rel_i: The relevance score of the keyphrase at rank i in the predicted ranking.
- $Irel_i$: the relevance score of the keyphrase at rank i in the ideal ranking.
- $log_2(i+1)$: logarithmic discount factor.

Lexical and rank-based evaluation metrics often overlook the semantic comparison of keyphrases. This challenge becomes even more pronounced with absent keyphrases, which may be more semantically accurate but differ literally from the reference keyphrases. Therefore, the study of semantic metrics is essential.

C. Semantic Similarity Metrics

Semantic similarity metrics represent the second generation of evaluation metrics, developed to overcome the limitations of lexical and rank-based metrics. Semantic similarity metrics represente predicted and reference keyphrases in a semantic space using pre-trained models such as BERT, RoBERTa, and Sentence-BERT, where the similarity between phrases is measured based on their semantic context. Prediction precision is calculated using cosine similarity provided by Formula 11.

$$\mathrm{Sim(PK,RK)} = \frac{\mathrm{e(PK).e(RK)}}{\|\mathrm{e(PK)}\|\|\mathrm{e(RK)}\|} \quad (11)$$

where e(·) is contextual embedding of a phrase obtained from a pretrained language model. Although this metric achieves semantic comparison, its effectiveness in keyphrase evaluation is limited because it compares the similarity of the phrase as a single unit, while lacking the ability to understand the word-for-word relationship between the prediction and the reference. To overcome this challenge, the BERTScore was proposed, which relies on contextual embedding that change depending on the word's position in the phrase. Instead to measure the cosine similarity between phrase vectors, the BERTScore aligns each token in the predicted phrase with the most semantically similar token in the reference keyphrase based on their context, thus preserving the value of each token in the keyphrase. To calculate BERTScore, precision and recall are first calculated by formulas 12 and 13.

$$\mathrm{Precision_{BERT}} = \frac{1}{|\mathrm{PK}|}\sum_{\mathrm{i=1}}^{|\mathrm{PK}|}\max_{\mathrm{j}}\cos\left(\mathrm{e(x_i)},\mathrm{e(y_j)}\right) \quad (12)$$

$$\mathrm{Recall_{BERT}} = \frac{1}{|\mathrm{RK}|}\sum_{\mathrm{j=1}}^{|\mathrm{RK}|}\max_{\mathrm{i}}\cos\left(\mathrm{e(y_j)},\mathrm{e(x_i)}\right) \quad (13)$$

where:

- e(·): The contextual embedding of a token obtained from BERT model
- cos(·): cosine similarity
- PK: Predicted keyphrase of tokens $\{x_1, \dots x_i, \dots x_{|PK|}\}$
- RK: Reference keyphrase of tokens $\{y_1, \dots y_j, \dots y_{|RK|}\}$

To calculate BERTScore, $Precision_{BERT}$ and $Recall_{BERT}$ combined using the harmonic mean via formula 14.

$$\mathrm{BERTScore} = 2\times\frac{\mathrm{Precision_{BERT}}\times\mathrm{Recall_{BERT}}}{\mathrm{Precision_{BERT}}+\mathrm{Recall_{BERT}}} \quad (14)$$

Instead of combining the words of a phrase into a single vector, BERTScore compares them word by word. Therefore, it addresses the problem of variations in phrase lengths. Also,

BERTScore evaluates semantic overlap, making it robust to paraphrasing in generative keyphrase prediction. In contrast, the precision of BERTScore is affected by the quality and comprehensiveness of the Gold Standard identified by human experts.

D. Human Reference Challenge

Manual identification keyphrases is a subjective cognitive task, where different keyphrases are selected for the same document based on the expert's background knowledge. The study [36], confirms that the manual selection of keyphrases is highly subjective. This makes relying on a single human reference limit the evaluation precision [12]. Fig. 2 illustrates the divergence between the keyphrases selected by author, Expert 1, and Expert 2.

Fig. 2. The divergence of reference keyphrases selection.

The central intersection represents the keyphrases agreed by everyone. Overlapping areas indicate partial agreement between two of them, while divergence areas represent keyphrases chosen by each independently. This representation highlights the inconsistency on gold standards to select keyphrases.

The cognitive effort expended in the manual selection, limits the number of keyphrases selected, this also, impact the evaluation process. Despite advancements in lexical and semantic evaluation metrics, the evaluation of keyphrase prediction models relies on the quality of the reference keyphrases. This leads to unreliable evaluation, especially in generative models, where they may generate semantically relevant keyphrases that are not present in the reference set.

There are several challenges to evaluate keyphrase prediction models. Current evaluation metrics remained to compare phrase-to-phrase, whether through exact or semantic matching. The keyphrases generated often represent a condensation of entire paragraphs, rather than a corresponding phrase in a human reference, which is often limited, and subjective. Therefore, we need a more comprehensive evaluation process. For this, we propose a new evaluation metric, instead of comparing predicted keyphrases with limited reference keyphrases, we compare the overall semantic impact of these phrases to generate the document summary, thus freeing the evaluation from limited human reference lists.

IV. Functional Evaluation Metric

A. Motivation for Functional Evaluation

The review of evaluation process for keyphrase prediction models in the previous section showed several challenges that limit the reliability of their results. Current evaluation metrics judge model performance solely by lexical or semantic matching between reference keyphrases and predicted keyphrases, neglecting characteristics such as topic coverage, diversity, and functional performance. Furthermore, they relie on the list of reference keyphrases, which are subjective and incomplete. Consequently, the performance of keyphrase prediction models is measured by human judgment rather than functional effectiveness, undermining the credibility of the evaluation process and penalizing models that generate functionally useful keyphrases but are not present in the reference list.

Therefore, we propose a Functional Evaluation Metric (FEF) that evaluates predicted keyphrases as a collective semantic unit, more independently of the reference list. FEF evaluates the performance of keyphrase prediction models by measuring the effectiveness of predicted keyphrases on two tracks. Abstract reconstruction track and text retrieval track, without relying on reference keyphrases. Therefore, the evaluation process is based on the functional success of the predicted keyphrases. We no longer measure whether this phrase matches the reference, but rather whether this phrase enables the system to summarize or retrieve the document.

B. Overview of FEM-KP

The Functional Evaluation Metric for KeyPhrase Prediction (FEM-KP) is a new metric designed to evaluate keyphrase prediction models based on the functional utility of the generated keyphrases, rather than measuring lexical or semantic similarity. FEM-KP measures the functional utility of generated keyphrases through two functions:

- Summary Reconstruction: This dimension measures the ability of keyphrases to serve as semantic seeds that can summarize the document and cover its content.
- Retrieval Power: This dimension measures the ability of keyphrases to identify the document within a large corpus.

The combinition of these dimensions in FEM-KP provides a uniform score that reflects the performance of the keyphrase prediction model to generate exploitable keyphrases for NLP applications such as text clustering, information retrieval, and document summarization.

C. Process of FEM-KP

To evaluate the performance of the keyphrase prediction model, the evaluation process consists of three main stages: synthesis, parallel evaluation tasks, and weight fusion. Fig. 3 illustrates how the FEM-KP score is calculated.

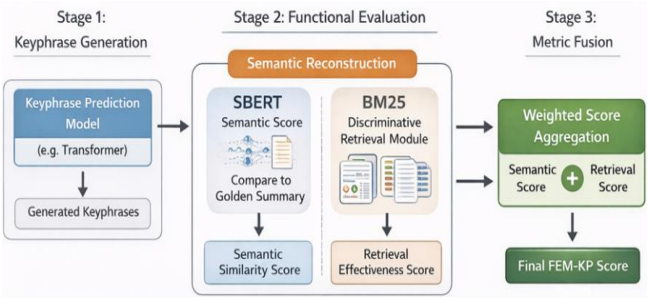

Fig. 3. Process of FEM-KP metric.

In the first stage, we use a transformer-based generator to combine and synthesize the generated keyphrases KP into a coherent semantic string SK. To achieve this, we first use a pre-trained generative model such as Text-to-Text Transfer Transformer (T5) [37] or Bidirectional and Auto-Regressive Transformers (BART) [25] for keyphrases-to-text conversion tasks.

To evaluate the ability of keyphrases to perform real tasks, FEM-KP across two parallel functional tracks. The first is the summative reconstruction score S_{summ}. In this track, we measure the semantic distance between the reconstructed summary SK and the gold summary GS using cosine similarity in the embedding space.

In contrast, the second track evaluates the ability of key phrases to recognize and retrieve the document D within a large dataset C. Its ability to handle term frequency values and normalize document length allows BM25 to calculate the retrieval score, which is an ideal metric to evaluate the retrieval utility of generated keyphrases.

The functional dimensions are combined into a single score to evaluate the overall quality of the keyphrases. The final FEM-KP score is calculated by balancing S_{summ} and S_{ret}.

D. Mathematical Formulation of FEM-KP Metric

FEM-KP framwork evaluate the quality of a keyphrase by measuring its functional utility across two NLP tasks. Let D is the source document, and $KP = \{P_1, P_2, \dots, P_n\}$ is the set of generated keyphrases from D by a keyphrase prediction model, and GS is the gold summary of D. The formulation follows a three-step pipeline:

1) Semantic reconstruction: Semantic reconstruction aims to generate a summary of the document SK based on keyphrases. T5 achieved a relative improvement in precision, but its performance is lower than BART in most summarization scenarios. For this, BART was used as a generative model in FEM-KP. To obtain the SK summary, we use the formula 15.

$$SK = arg\ \max_{s} p_\theta(s|Serialize(KP)) \quad (15)$$

where:

- SK: The generated summary.
- p_θ: The conditional probability modeled by BART
- Serialize(KP): A linearized representation of the KP set.
- s: A candidate output text sequence generated by the BART decoder.

The generate summary SK obtained from BART is represented using a Sentence-BERT [38]. The vector obtained serves as a semantic fingerprint for the generated keyphrases.

The first Track Functional Evaluation, is the ability of keyphrases to generate a document summary. In this track, we measure the semantic distance S_{summ} in vector space between the generate summary SK and the gold summary GS. To calculate semantic similarity, the SK and GS are converted into semantically vectors using SBERT model in formula 16.

$$\begin{cases} V_k = Pooling(Transformer_\theta(SK)) \\ V_g = Pooling(Transformer_\theta(GS)) \end{cases} \quad (16)$$

S_{summ} is calculated using cosine similarity in formula 17.

$$S_{summ} = \frac{\sum_{i=1}^{n} v_{k,i} \cdot v_{g,i}}{\sqrt{\sum_{i=1}^{n} v_{k,i}^2}\sqrt{\sum_{i=1}^{n} v_{g,i}^2}} \quad (17)$$

We used KP20k [39], Inspec [40], and Krapivin [41] as benchmark datasets for comparison. Their features and the reasons for their selection will be explained in the evaluation section.

2) Retrieval rank: The second track evaluates the ability of keyphrases to identify the source document D from a large set of documents C. To calculate the retrieval score, we use the BM25 probabilistic model, which improves the traditional TF-IDF model by incorporating term frequency saturation and document length normalization. Formula 18 calculates the document retrieval rank D, using the keyphrases set K.

$$Rank_K(D) = \sum_{p \in K} IDF(p).\frac{fr(p,D).(n+1)}{fr(p,D)+n.(1-\alpha+\alpha.\frac{|D|}{Avg(C)})} \quad (18)$$

where:

- $IDF(p)$: The Inverse Document Frequency of keyphrase p.
- $fr(p,D)$: The frequency of keyphrase p within document D.
- $|D|$: The length of the D.
- $Avg(C)$: The average document length across the corpus C.
- n: A hyperparameter to control the term frequency saturation. (Typically, between 1.2 and 2).
- α: A scaling factor for document length normalization. (typically, α =0.75).

It is difficult to combine the rank as score with the S_{summ} score (values between 0 and 1) in a single equation, for this normmalization is necessary. The S_{ret} score is calculated as the reciprocal rank. The retrieval score is calculate using formula 19.

$$S_{ret} = \begin{cases} \frac{1}{Rank_K(D)}, & if\ Rank_K(D) \le 100 \\ 0, & otherwise \end{cases} \quad (19)$$

The retrieval function score approaches zero if the keyphrases fail to rank the document among the top 100 results. This significantly speeds up the evaluation process, especially in datasets containing thousands of documents.

The functional dimension scores are combined into a single score to evaluate the overall performance of keyphrases. The FEM-KP score is calculated by balancing the sum of the S_{summ} and S_{ret}. This allows to adjust the score based on application priority, where a higher β Hyperparameter is used for summarization tasks, and a higher δ Hyperparameters for search optimization or indexation tasks. The final FEM-KP score was calculated using the formula 20.

$$FEF - KP = \beta . S_{summary} + \delta . S_{retrieval} \quad (20)$$

β and δ: Hyperparameters that are controlled according to the nature of the task, often $\beta + \delta = 1$.

This formula includes the effectiveness evaluation of the generated keyphrases, especially in the semantic representation of the document, and retrieval tasks.

E. Advantages and Limitations

Instead of the lexical or semantic matching used by current evaluation metrics, the FEM-KP metric adopts functional evaluation, which measures the effectiveness of keyphrases by their application in NLP tasks. The FEM-KP metric is characterized by its flexibility in adjusting the values of the hyperparameters β and γ, which allowed for the selection of the most suitable model for each task. Also, using the BM25 algorithm, the keyphrase prediction model could not achieve a high score when generating a large number of general keyphrases.

Despite these advantages, the quality of the reference summaries used may affect the reliability of the performance. Furthermore, they may not always be available. Also, using pre-trained language models and retrieval algorithms increases the computational cost of FEM-KP compared to simple lexical metrics. Also, the sensitivity of the parameters requires careful selection of β and γ values to avoid biased results

V. Experimental Setup and Results

This section details the implementation procedures and evaluation protocol for FEM-KP, including evaluation datasets and keyphrase prediction models. This section aims to ensure a comprehensive evaluation that allows for comparison of FEM-KP with current evaluation metrics.

A. Experimental Setup

1) Evaluation protocol: The evaluation protocol involves comparing the performance of key phrase prediction models using various evaluation metrics, including functional effectiveness. This comparison is achieved by calculating the BERTScore, ROUGE, and F1Measure scores for each model and comparing them to the FEM-KP scores for each model.

2) Evaluation datasets: To evaluate the FEM-KP, we rely on three reference evaluation datasets. The KP20k dataset comprises scientific abstracts from multiple disciplines. Each document has keyphrases defined by author. The size and diversity of KP20k content make it suitable for comparisons between different evaluation metrics. The second dataset, is Inspec. It smaller but more carefully curated and contains scientific abstracts. these datasets have been widely used in previous studies, making them suitable for comparing the FEM-KP with exact-match and semantic-based evaluation metrics.

TABLE II. Comparison of the Datasets Used for FEM-KP Evaluation

Dataset	Domain	# Test Documents	Avg. Text Length	Keyphrase Types
KP20k	Scientific Abstracts	20,000	180 words	Present & Absent
Inspec	Scientific Abstracts	500	135 words	Present & Absent
Krapivin	Scientific Papers	2300	8000 words	Present & Absent

To evaluate FEM-KP in long documents and complex retrieval scenarios, we exploited the Krapivin dataset, which includes scientific papers. Table II presents the features of KP20k and Inspec datasets.

3) Keyphrase prediction models: For a comprehensive comparison of the evaluation metrics, TextRank was adopted as an unsupervised, graph-based model that extracts keyphrases from the word-to-word structure within the text. CopyRNN was adopted as a supervised, recurrent neural network model that utilizes a copy mechanism to generate present and absent keyphrases in the text. Additionally, BART-based Keyphrase Generator and T5-based Keyphrase Generator were adopted, leveraging of pre-trained generative models to produce high-quality, sequential keyphrases. The diversity of models including unsupervised, supervised, and generative models, allows to evaluate FEM-KP in different scenarios. These models represent the evolution of keyphrase prediction task. Table III provides a comprehensive analysis of each model type and the rationale for its selection.

To highlight the performance of FEM-KP across different datasets for the four models and compare it to the F1-Score, ROUGE-L and BERTScore metrics. Fig. 4 summarizes the different stages of experimental evaluation.

TABLE III. A COMPREHENSIVE ANALYSIS OF KEYPHRASE PREDICTION MODELS USED IN EVALUATION

Model	Model Type	Output	Reason
TextRank [42]	Unsupervised, Graph-based	Extractive keyphrases	An unsupervised model to extract keyphrase.
CopyRNN [39]	Supervised, Neural Seq2Seq	Extractive and generative keyphrases	A neural network model to generate present and absent keyphrases.
BART-based Keyphrase Generator [25]	Transformer-based	Generative keyphrases	Abstractive model used for keyphrases prediction.
T5-based Keyphrase Generator [37]	Transformer-based (Text-to-Text)	Generative keyphrases	A metric for text transformation. Widely used in recent studies to generate keyphrases.

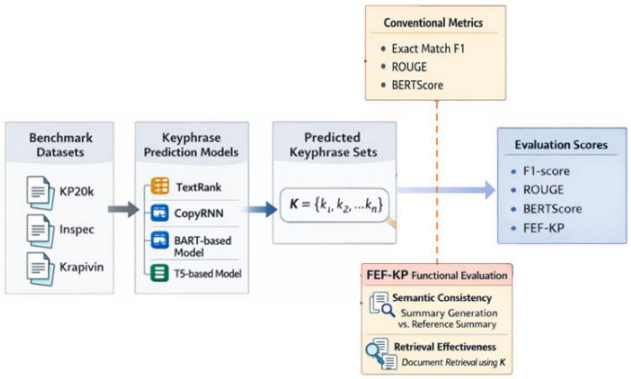

Fig. 4. The stages of experimental evaluation

We will present, analyze and discuss the evaluation results in the following part.

B. Evaluation Results

1) Sample size: To ensure the reliability of comparisons between the four models and the metrics used, samples were selected from the KP20k, Inspec, and Krapivin datasets. Table IV shows the sample size used for each dataset.

TABLE IV. THE SAMPLE SIZE USED IN EVALUATION

Dataset	Sample Size	Avg. Document Length
Inspec	500	Short (~120 words)
KP20k	2,000	Medium (~180 words)
Krapivin	400	Long (~8,000+ words)

To reduce computational costs and achieve a high level of reliability, 2,000 documents were selected from the KP20k dataset, and we used all documents from the Inspec dataset. 400 research papers were selected from the Krapivin dataset to evaluate the effectiveness retrieval of the FEM-KP in longer documents.

2) Consistency of performance ranking: The consistent ranking of models across diverse datasets is a prerequisite for any evaluation. The reliability of an evaluation metric is measured by its ability to rank model performance across diverse datasets. Despite the complexity of the dataset or the length of the document, the relative ranking of models should remain constant. In this part, we analyze the robustness of the FEM-KP metric by comparing its ranking stability for key phrase prediction models with the F1-Score, ROUGE-L, and BERTScore metrics. Our analysis aims to demonstrate that FEM-KP has a unique ability to identify the most effective model even as data complexity increases, whereas other measures are susceptible to ranking inversions and contextual noise. Table V presents the performance of the four keyphrases prediction in the Inspec dataset representing short texts.

TABLE V. THE PERFORMANCE OF THE FOUR KEYPHRASES PREDICTION IN THE INSPEC DATASET

Model	F1-Score	ROUGE-L	BERTScore	FEM-KP
TextRank	(4) 0.215	(4) 0.312	(4) 0.721	(4) 0.584
CopyRNN	(3) 0.248	(3) 0.345	(3) 0.765	(3) 0.642
T5-based	(2) 0.278	(2) 0.372	(2) 0.835	(2) 0.775
BART-based	(1) 0.285	(1) 0.380	(1) 0.842	(1) 0.791

The texts in Inspec are short summaries, for this, the potential error across all metrics is reduced. Therefore, we observe consistency across all metrics in ranking model performance. Conversely, when using the KP20K dataset, which contains longer summaries compared to Inspec, the exact Match metrics began to deviate. F1-Score favored CopyRNN over T5 and ROUGE-L favored T5 over BART, while BERTScore and FEM-KP maintained their rankings. Table VI presents the performance of the four keyphrases prediction in the KP20k dataset representing short texts.

TABLE VI. THE PERFORMANCE OF THE FOUR KEYPHRASES PREDICTION IN THE KP20K DATASET

Model	F1-Score	ROUGE-L	BERTScore	FEM-KP
TextRank	(4) 0.142	(4) 0.245	(4) 0.680	(4) 0.510
CopyRNN	**(2) 0.200**	(3) 0.290	(3) 0.730	(3) 0.615
T5-based	**(3) 0.185**	**(1) 0.325**	(2) 0.808	(2) 0.732
BART-based	(1) 0.210	**(2) 0.318**	(1) 0.815	(1) 0.748

The deviate in the ranking of exact matching metrics is due to the increasing size and diversity of the data. Additionally, these metrics are sensitive to the Longest Common String (LCS). The results also demonstrate that FEM-KP overcomes this problem through its reconstruction track. The greatest challenge arises, in Krapivin, as the metric must evaluate the performance of models in long texts. Table VII presents the performance of the four keyphrases prediction in the Krapivin dataset representing long texts.

In Krapivin, a deviate is observed in the F1-Score, where CopyRNN was favored over T5. This is due to the length of the texts, which increased the probability of random matches. Furthermore, the semantic-based BERTScore metric showed ranking instability within the Krapivin dataset. The length of

the texts also led to contextual noise resulting from the presence of irrelevant informations. Therefore, CopyRNN obtiened a high score by BERTScore because the context was perfectly matched. In contrast, FEM-KP is the only metric that maintained the rank (BART > T5 > CopyRNN > TextRank) in all datasets, proving that it is not fooled by length of the text or overlapping of phrases, but rather measures functional value.

TABLE VII. THE PERFORMANCE OF THE FOUR KEYPHRASES PREDICTION IN THE KRAPIVIN DATASET

Model	F1-Score	ROUGE-L	BERTScore	FEM-KP
TextRank	(4) 0.125	(4) 0.210	(4) 0.655	(4) 0.495
CopyRNN	**(2)0.180**	(3) 0.265	**(2) 0.785**	(3) 0.580
T5-based	**(3) 0.175**	(2) 0.302	**(3) 0.782**	(2) 0.695
BART-based	(1) 0.195	(1) 0.310	(1) 0.790	(1) 0.712

3) Discussion of results: Analysis of results from the Inspec, KP20K, and Krapivn datasets revealed differences between match evaluation and functional evaluation. While F1-Score, ROUGE-L, and BERTScore metrics showed deviate ranking with increasing document complexity and length, with extractive models like CopyRNN outperforming generative models. They often rewarded lexical redundancy and contextual matching at the expense of semantic utility. In contrast, FEM-KP demonstrated ranking consistency (BART > T5 > CopyRNN > TextRank) regardless of datasets. This stability is attributed to the dual-tracks structure of our metric. The reconstruction track rewards only the keyphrases capable to generate the document maining, while the retrieval track rewards those capable to retrieve the document. The FEM-KP result proves to be unaffected by text length and independent of the model. This provide a more reliable metric to evaluate generative keyphrase prediction models. These results confirm that the FEM-KP not only measures similarity but also functional information, making it a more reliable tool for researchers in the NLP field.

VI. CONCLUSION

In this study, we addressed the evaluation of keyphrase prediction models by highlighting the challenges posed by current evaluation metrics such as F1-Score, ROUGE-L, and BERTScore, which primarily focus on exact matching or contextual similarity but often fail to detect the functional utility of keyphrases. This leads to inconsistent model rankings, particularly in long and complex documents. Therefore, to overcome this problem, we proposed the FEM-KP metric. Our metric adopts a two-track evaluation, the utility of reconstruction (track A) and semantic consistency (track B), which has allowed to distinguish between contextually similar phrases and functionally necessary keyphrases. Our experimental results across three diverse datasets, Inspec, KP20k, and Krapivin showed that FEM-KP provides a superior and more stable evaluation metric and provide a fair assessment that rewards the functional performance of the phrases rather than literal repetition compared to current evaluation metrics. In conclusion, instead to evaluate keyphrase prediction models based to compare the predicted keyphrases with the reference keyphrases, FEM-KP allows to evaluate these models based on the functional performance of these phrases. Therefore, we believe this metric will provide a more reliable benchmark for researchers to develop highly effective NLP systems. Therefore, we will expand FEM-KP in the future to include other functions such as classification and clustering.

REFERENCES

[1] E. Papagiannopoulou and G. Tsoumakas, "A review of keyphrase extraction," WIREs Data Min & Knowl, vol. 10, no. 2, p. e1339, Mar. 2020, doi: 10.1002/widm.1339.

[2] C. Yoo and H. Lee, "Improving Abstractive Dialogue Summarization Using Keyword Extraction," Applied Sciences, vol. 13, no. 17, p. 9771, Aug. 2023, doi: 10.3390/app13179771.

[3] F. Boudin, Y. Gallina, and A. Aizawa, "Keyphrase Generation for Scientific Document Retrieval," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1118–1126. doi: 10.18653/v1/2020.acl-main.105.

[4] J. Gabín, J. Parapar, and C. Macdonald, "Beyond Questions: Leveraging ColBERT for Keyphrase Search," 2024, arXiv. doi: 10.48550/ARXIV.2412.03193.

[5] L. T. Do, A. Bodke, P. S. Akash, and K. C.-C. Chang, "ERU-KG: Efficient Reference-aligned Unsupervised Keyphrase Generation," 2025, arXiv. doi: 10.48550/ARXIV.2505.24219.

[6] D. Wu, D. Yin, and K.-W. Chang, "KPEval: Towards Fine-Grained Semantic-Based Keyphrase Evaluation," 2023, arXiv. doi: 10.48550/ARXIV.2303.15422.

[7] H. Ding and X. Luo, "LongDocRank: graph-augmented large language models for unsupervised keyphrase extraction from long documents," J Big Data, Jan. 2026, doi: 10.1186/s40537-025-01339-8.

[8] Y. Luo, Y. Xu, J. Ye, X. Qiu, and Q. Zhang, "Keyphrase Generation with Fine-Grained Evaluation-Guided Reinforcement Learning," in Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 2021, pp. 497–507. doi: 10.18653/v1/2021.findings-emnlp.45.

[9] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," 2019, arXiv. doi: 10.48550/ARXIV.1904.09675.

[10] J. Jose and B. Soundarabai, "A Survey on Machine Learning Based Keyphrase Generation in Natural Language Processing," Nepal Journal of Science and Technology, vol. 22, no. 2, pp. 66–74, Dec. 2023, doi: 10.3126/njst.v22i2.85238.

[11] L. Ajallouda, F. Z. Fagroud, A. Zellou, and E. H. Benlahmar, "A Systematic Literature Review of Keyphrases Extraction Approaches," Int. J. Interact. Mob. Technol., vol. 16, no. 16, pp. 31–58, Aug. 2022, doi: 10.3991/ijim.v16i16.33081.

[12] K. S. Hasan and V. Ng, "Automatic Keyphrase Extraction: A Survey of the State of the Art," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, 2014, pp. 1262–1273. doi: 10.3115/v1/P14-1119.

[13] L. Ajallouda, F. Z. Fagroud, A. Zellou, and E. H. Benlahmar, "Automatic keyphrases extraction: an overview of deep learning approaches," Bulletin EEI, vol. 12, no. 1, pp. 303–313, Feb. 2023, doi: 10.11591/eei.v12i1.4130.

[14] M. Song, Y. Feng, and L. Jing, "A Survey on Recent Advances in Keyphrase Extraction from Pre-trained Language Models," in Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, 2023, pp. 2153–2164. doi: 10.18653/v1/2023.findings-eacl.161.

[15] B. Xie et al., "From Statistical Methods to Deep Learning, Automatic Keyphrase Prediction: A Survey," 2023, doi: 10.48550/ARXIV.2305.02579.

[16] N. Giarelis and N. Karacapilidis, "Deep learning and embeddings-based approaches for keyphrase extraction: a literature review," Knowl Inf Syst, vol. 66, no. 11, pp. 6493–6526, Nov. 2024, doi: 10.1007/s10115-024-02164-w.

[17] M. Umair, T. Sultana, and Y.-K. Lee, "Pre-Trained Language Models for Keyphrase Prediction: A Review," 2024, doi: 10.48550/ARXIV.2409.01087.

[18] B. Kang and Y. Shin, "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.

[19] H. Cao, "Recent advances in text embedding: A Comprehensive Review of Top-Performing Methods on the MTEB Benchmark," 2024, arXiv. doi: 10.48550/ARXIV.2406.01607.

[20] A. Urlana, P. Mishra, T. Roy, and R. Mishra, "Controllable Text Summarization: Unraveling Challenges, Approaches, and Prospects - A Survey," in Findings of the Association for Computational Linguistics ACL 2024, Bangkok, Thailand, 2024, pp. 1603–1623. doi: 10.18653/v1/2024.findings-acl.93.

[21] G. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," jair, vol. 22, pp. 457–479, Dec. 2004, doi: 10.1613/jair.1523.

[22] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," 2015, arXiv. doi: 10.48550/ARXIV.1509.00685.

[23] R. Nallapati, B. Zhou, C. Dos Santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," in Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 2016, pp. 280–290. doi: 10.18653/v1/K16-1028.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018, arXiv. doi: 10.48550/ARXIV.1810.04805.

[25] M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," 2019, arXiv. doi: 10.48550/ARXIV.1910.13461.

[26] T. B. Brown et al., "Language Models are Few-Shot Learners," 2020, arXiv. doi: 10.48550/ARXIV.2005.14165.

[27] Y. Zhang, H. Jin, D. Meng, J. Wang, and J. Tan, "A comprehensive survey on automatic text summarization with exploration of LLM-based methods," Neurocomputing, vol. 663, p. 131928, Jan. 2026, doi: 10.1016/j.neucom.2025.131928.

[28] S. Mushtaq and K. Veningston, "Biomedical text summarization with large language models: methodologies, challenges, and future directions," Int J Data Sci Anal, vol. 22, no. 1, p. 29, Dec. 2026, doi: 10.1007/s41060-025-00956-z.

[29] L. Bednarczyk et al., "Scientific Evidence for Clinical Text Summarization Using Large Language Models: Scoping Review," J Med Internet Res, vol. 27, p. e68998, May 2025, doi: 10.2196/68998.

[30] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," FNT in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/1500000019.

[31] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," 2020, arXiv. doi: 10.48550/ARXIV.2004.04906.

[32] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," 2020, arXiv. doi: 10.48550/ARXIV.2004.12832.

[33] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise Zero-Shot Dense Retrieval without Relevance Labels," in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023, pp. 1762–1777. doi: 10.18653/v1/2023.acl-long.99.

[34] W. X. Zhao, J. Liu, R. Ren, and J.-R. Wen, "Dense Text Retrieval based on Pretrained Language Models: A Survey," 2022, arXiv. doi: 10.48550/ARXIV.2211.14876.

[35] Y.-A. Liu, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng, "Robust Neural Information Retrieval: An Adversarial and Out-of-distribution Perspective," 2024, arXiv. doi: 10.48550/ARXIV.2407.06992.

[36] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles," Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 21–26, 2010.

[37] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," 2019, arXiv. doi: 10.48550/ARXIV.1910.10683.

[38] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019, pp. 3980–3990. doi: 10.18653/v1/D19-1410.

[39] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep Keyphrase Generation," 2017, arXiv. doi: 10.48550/ARXIV.1704.06879.

[40] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 216–223. doi: 10.3115/1119355.1119383

[41] M. Krapivin, A. Autayeu, and M. Marchese, "Large dataset for keyphrases extraction," Technical Report, University of Trento, 2010.

[42] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, Barcelona, Spain, 2004, pp. 20-es. doi: 10.3115/1219044.1219064.