Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 19 No. 21 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijim.v19i21.56747

PAPER

Pattern Prediction: A Comprehensive Review of Evaluation Methods

Amine Berquedich[1], Lahbib Ajallouda[2] (✉), Ahmed Zellou[1], Inès Chihi[3]

[1]Mohammed V University, Rabat, Morocco

[2]Ibn Zohr University, Agadir, Morocco

[3]University of Luxembourg, Luxembourg City, Luxembourg

l.ajallouda@uiz.ac.ma

ABSTRACT

Keyphrases are expressions allow to identify the topics, or the ideas addressed in a document. Various natural language processing (NLP) applications have employed these phrases to improve their performance. Several research papers have been published in recent years that focus on keyphrases extraction or generation approaches. Various techniques are used to evaluate the performance of these methods. These techniques can be classified into three main categories: lexical similarity, rank-based, and semantic similarity techniques. This paper presents a comprehensive review of these techniques, with a particular focus on the major challenges encountered during the evaluation process. Also, this review underscores the need for more robust and rigorous evaluation techniques to improve the reliability and effectiveness of keyphrases prediction approaches. This review will contribute to understanding the different keyphrases evaluation techniques, the main challenges they face, and future research directions related to keyphrases prediction evaluation.

KEYWORDS

natural language processing (NLP), Keyphrase prediction approaches, Keyphrase evaluation challenges, Keyphrases evaluation techniques

1 INTRODUCTION

The huge amount of text data produced daily makes it very difficult to analyze. Therefore, to identify the main topics and ideas of a text, it is necessary to find smart solutions. Keyphrases are one of the solutions used to improve the performance of natural language processing (NLP) applications [1]. Keyphrases prediction in a document is based on two mechanisms, extraction and generation [2]. Keyphrases are extracted either using supervised models based on machine and deep learning algorithms, or using unsupervised methods based on statistics, graphs and sentence embedding techniques [3]. These methods are commonly evaluated using lexical similarity measures, rank-based comparisons, and semantic similarity techniques.

Berquedich, A., Ajallouda, L., Zellou, A., Chihi, I. (2025). Pattern Prediction: A Comprehensive Review of Evaluation Methods. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(21), pp. 122–144. https://doi.org/10.3991/ijim.v19i21.56747

Article submitted 2025-05-20. Revision uploaded 2025-08-17. Final acceptance 2025-08-17.

However, each of these evaluation techniques faces challenges that can affect the accuracy and consistency of the results.

This paper aims to review the techniques and processes currently used to evaluate keyphrase extraction or generation methods. The review seeks not only to classify and describe the various evaluation techniques, but also to critically examine their strengths, weaknesses, and applicability across different contexts. This review serves as a foundation for identifying directions for future research in the keyphrase prediction field, including the development of effective assessment protocols that emulate human evaluation.

The remainder of this paper will include a section introducing the most important keyphrase prediction methods. Section 3 will analyze the evaluation of keyphrase prediction models. Section 4 will discuss the results of this comprehensive review and the challenges posed by the current evaluation of these models. The paper will conclude by proposing future research directions to overcome the current challenges.

2 KEYPHRASES PREDICTION MODELS

Keyphrases prediction, whether using extraction or generation mechanisms, is a natural language processing task used in many applications such as text summarization, information retrieval, text clustering, and recommendation systems. Many studies [2], [4], [5], [6] classify keyphrases into present keyphrases that appear in the document and absent keyphrases that are automatically generated. In this section we will present the most important approaches of keyphrases prediction, which adopted the extraction mechanism or the generation mechanism.

2.1 Keyphrases extraction approaches

The keyphrase extraction task is to identify the phrases in the document that express its main content [7]. Several works have concerned a review of keyphrase extraction [8] focused on errors during the evaluating keyphrase extraction process. While [9] investigated the effect of text preprocessing on key phrase extraction. [10] examined different types of features, training and evaluation datasets associated with keyphrases extraction. The authors of [11] gave a general introduction to the field of keyphrases extraction by analyzing the advantages and disadvantages of keyphrases extraction methods. We also accomplished in [3] a systematic literature review of keyphrases extraction methods published between 2015 and 2022.

- **Keyphrases extraction process:** Several methods have been proposed to extract keyphrases, supervised and unsupervised. According to previous studies [10], [12], supervised methods perform better than unsupervised methods. However, they remain less popular because they require provide training data. We will introduce the various techniques used in each category, as well as the evaluation measures and datasets used for training and evaluation.

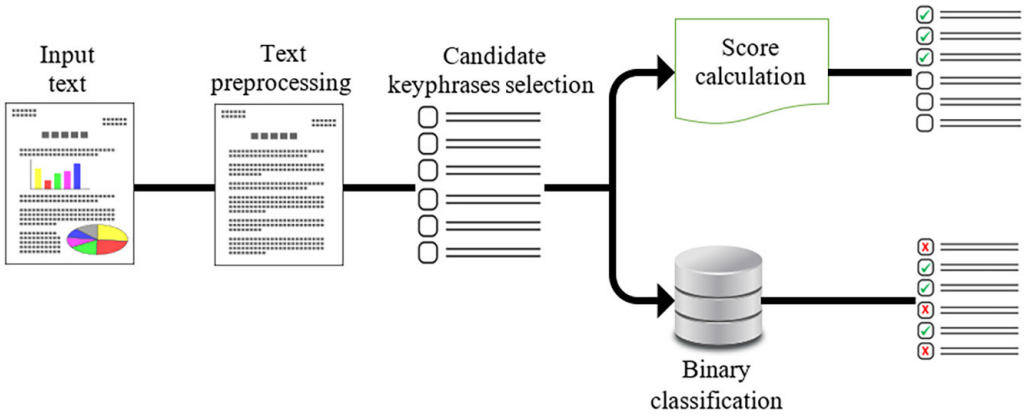

Fig. 1. The keyphrases extraction process

The keyphrases extraction process involves several stages [13]. Document preprocessing is the first step in which non-text data and stop words are removed [14]. The second step aims to identify the set of candidate phrases. In the third stage, the keyphrases are selected from the candidate phrases set. To select these phrases, the principles of order or classification are adopted. In the first case a score is calculated for each candidate phrase. The second case requires the availability of a dataset for learning. Figure 1 present the stages of keyphrase extraction process.

- **Unsupervised approaches:** Unsupervised methods adopt the principle of ranking the set of candidates keyphrases, according to one or more weighting coefficients. The candidates keyphrases with a weight value greater than the minimum are the keyphrases set. PC = {P_1, P_2, ..., P_n} is the candidate keyphrases set of a document D. W(P_i) the weight of candidate P_i. The keyphrases set of document D, corresponds to the following formula (1):

$$S_{kp} = \bigcup_{P_i \in PC} \{P_i : W(P_i) > \min\} \quad (1)$$

Where: S_{kp} is the set of keyphrases from document d. According to several reviews [3], [10]. Unsupervised methods can be classified into three categories:

1. Methods based on statistical models: These methods exploit statistical features in the document to calculate the scores for candidate keyphrases [15], [16], [17].
2. Methods based on Graph: Most of these methods rely on the co-occurrence relationship between document phrases [18], [19] while other methods try to add phrase position, syntactic and semantic relationship between phrases to improve keyphrase extraction [20], [21].
3. Methods based on sentence embedding techniques: The candidate keyphrases and the document are represented based on one of the words or sentences embedding techniques [22], [23], [24].

Unsupervised keyphrase extraction methods exploit several techniques, such as statistic features, graph representation, words and sentences embedding. Generally, unsupervised methods follow the principle of ranking candidate keyphrases according to their importance degree for the document. Identifying the number of key phrases in a document remains a weak point for these methods. Despite this, most of these methods are not affected by the language of the document and do not require a corpus for training.

- **Supervised approaches:** Most supervised approaches learn to classify candidate phrases into "keyphrases" or "non-keyphrases." The keyphrases extraction via supervised approaches is treated as a classification problem [25] using a binary classification function (2):

$$f: \mathbb{R}^m \rightarrow \{1, -1\} \tag{2}$$

Where: f is a function that takes as argument a features vector $V(P_{ij}(f_1(P_{ij}), \ldots f_m(P_{ij}))$. f returns a value of 1 if the candidate P_{ij} is a keyphrase in document i and a value –1 if it is non-keyphrase. The keyphrases of a D_i document can be represented by formula (3).

$$M_i = \bigcup_{P_{ij} \in D_i} \left\{ P_{ij} : f(P_{ij}) = 1 \right\} \tag{3}$$

Generally, supervised methods require a large amount of training data. Providing this data requires a lot of effort. However, many supervised methods have been proposed. We have classified them according to the learning algorithm used.

  - Naive Bayes (NB) algorithm is a probabilistic machine learning model for binary classification. NB has been adopted by many keyphrases extraction methods such as KEA model [26], Nguyen model [27] and CRF model [28].
  - Support vector machine (SVM) algorithm is based on the creation of a decision boundary capable of separating the space into classes in order to place the data to be classified in the right category. This algorithm has been used in many methods to identify keyphrases [29], [30].
  - Neural network models are models used more in keyphrases generation, but we also find methods that used these techniques in keyphrases extraction, such as [31], [32].

The keyphrases extraction via supervised approach is a classification task. These approaches are affected by the domain of the document and the language in which it is written. This problem requires having a training corpus whenever the domain and language in which the document is written change. Redundancy in the extracted keyphrases is another problem with supervised methods.

2.2 Keyphrases generation approaches

Several studies [12], [33], [34] have confirmed that there are two keyphrases types, present and absent keyphrases. It is not possible to identify absent keyphrases using the extraction mechanism. The development of Seq2Seq techniques [35] proposed the KeyPhrase Generation (KPG) rather than their extraction. KPG is an automatic prediction of phrases that highlight important information in a document [36]. The importance of KPG also lies in their contribution to improving the performance of natural language processing tasks [37]. In the literature, many KPG methods have been proposed. According to several studies [2], [4], keyphrases are generated using three training models, One2One [12], One2Seq [38], and One2Set [39].

- **Training paradigms:** Three main models are used to train the KPG approaches, One2One, One2Seq and One2Set. These models are based on the Seq2Seq model. This paradigm has been described in [35], [40]. KPG is a process similar to

language prediction. Each keyphrase is a short sequence of tokens. To generate keyphrase based on Seq2seq model, the source text is used as the input sequence for the encoder which collects the important information in the context vector. The decoder uses this vector to generate multiple sequences of tokens, each sequence is a keyphrase. Figure 2 introduces the process of keyphrases generation based on Seq2seq model.

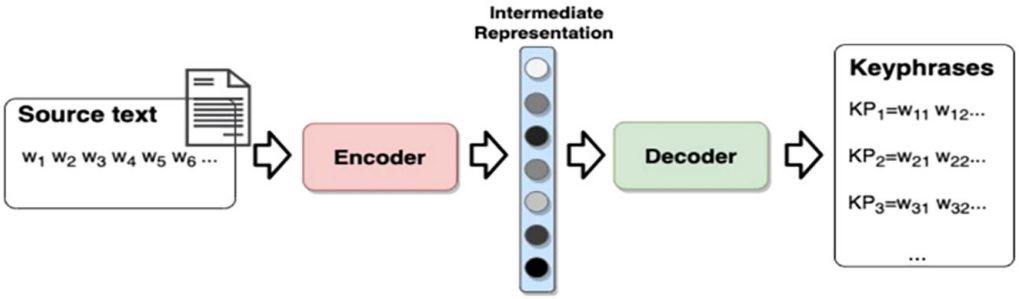

Fig. 2. Keyphrases generation process based on Seq2seq model

Most KG methods are based on the seq2seq model, which is exploited via a training model. The main difference between these models is how they handle multiple target keyphrases. To understand how keyphrases are generated, we will first present these models.

- **One2One model:** One2One [12] is the first training model proposed for KPG. The One2One training paradigm is based on compressing the overall source text content into a hidden context vector using an encoder. The decoder generates keyphrases based on the context vector. Generally, a training set consisting of N data. Each data sample $(X^{(i)}, P^{(i)})$ contains one source text $X^{(i)}$ and K_i target keyphrases $P^{(i)} = (p^{(i,1)}, p^{(i,2)}, ..., p^{(i,Ki)})$, where x(i) source text and the $p^{(i,j)}$ keyphrase are sequences of words:

$$\begin{cases} X^{(i)} = \left\{ x_1^{(i)}, x_2^{(i)}, \ldots, x_{L_{x^{(i)}}}^{(i)} \right\} \\ p^{(i,j)} = \left\{ w_1^{(i,j)}, w_2^{(i,j)}, \ldots, w_{L_{p^{(i,j)}}}^{(i,j)} \right\} \end{cases} \quad (4)$$

Where:

- $L_{x^{(i)}}$: The number of words in the sequence $X^{(i)}$.
- $L_{p^{(i,j)}}$: The number of words in the sequence $p^{(i,j)}$.

Each data simple $(X^{(i)}, P^{(i)})$ contains a set of target phrase sequences and a source text sequence. Using the RNN encoder-decoder requires the conversion of $(X^{(i)}, P^{(i)})$ into text-keyphrase pairs that contain only a source sequence and a one-phrase sequence. The split of the data sample $(X^{(i)}, P^{(i)})$ is represented by formula (5).

$$(X^{(i)}, P^{(i)})\{(X^{(i)}, P^{(i,1)}), (X^{(i)}, P^{(i,2)}), ..., (X^{(i)}, P^{(i,L)})\} \quad (5)$$

Where:

- L: The number of targets keyphrases in the data sample $(X^{(i)}, P^{(i)})$

After splitting the sample data, the Encoder-Decoder model is ready to learn the mapping from the source text sequence to the target sequence. Figure 3 shows an example.

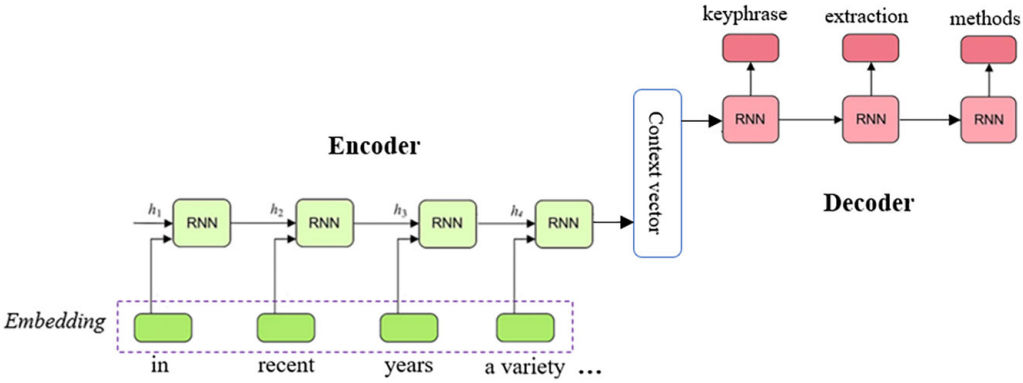

Fig. 3. An example of KPG according to the One2One training model

Methods that adopt the One2One training paradigm compress the overall source text content into a hidden context vector using an encoder. The decoder generates one keyphrase in each sequence. To improve the performance of KPG, some methods added other mechanisms during training. CorrRNN [41] uses the coverage and review mechanisms to cover all parts of the document. While ParaNet [42] used encoders to compress words and POS tags to overcome overlapping keyphrases being generated.

- **One2Seq model:** One2Seq [38] is the second proposed training model for KPG. Such as One2One, One2seq is based on the principle of Seq2Seq. To create the training set, the One2Seq model combines all p_i target keyphrases into a single sequence. P=<bos>p_1<sep> ... <sep>p_n<eos>. The sequence is prefixed by <bos> and terminated by <eos>. Between each two target phrases, a <sep> is inserted. Thus, only one data point (t; P) is formed. The One2Seq model was trained to predict a single sequence containing all target phrases. Which contributed to avoid generating similar keyphrases. Figure 4 provides an example of data point construction.

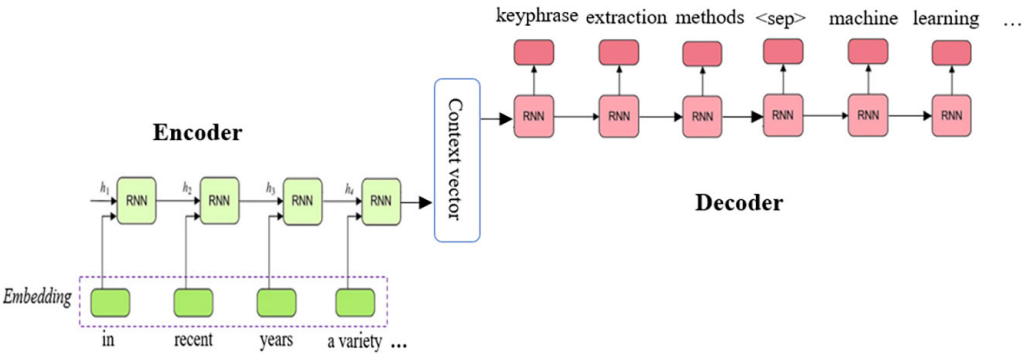

Fig. 4. An example of KPG according to the One2Seq training model

For the present keyphrases they are ordered in P sequence according to the order of their appearance in the source text. Absent keyphrases are appended at the end of the sequence. This helps the model to learn the dependencies between the keyphrases. Also, it specifies the number of target phrases in the source text. However, during training in the One2Seq model, the decoder takes a sequence of keyphrases as the target. Training to generate keyphrases in a single sequence is often affected by the order of these phrases in the sequence. This may create a false bias, which was confirmed by the empirical study published in [43].

- **One2Set Model:** The ONE2SET model [39] considers that training to generate one keyphrase at a time or a single sequence of keyphrases introduces a wrong bias. To resolve this problem, it treats the keyphrase generation as a set generation task. That is, predicting target keyphrases in parallel as a set as shown in Figure 5.

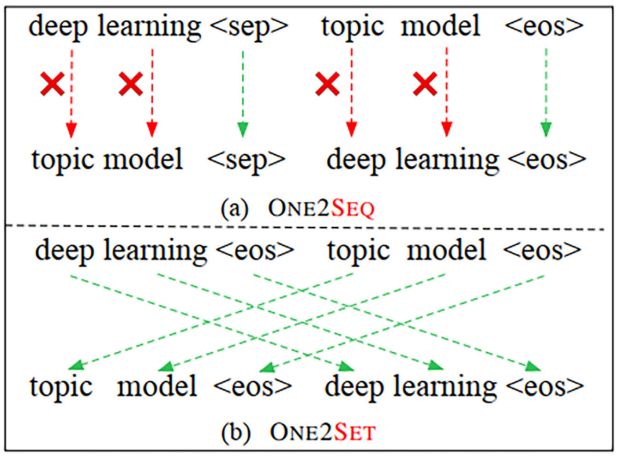

Fig. 5. An example comparing the KPG between ONE2SEQ and ONE2SET model [39]

The sample data used for training consisted of (t, Y) pairs. t represents the source text, and y represents a set of target keyphrases as indicated by the formula (6).

$$Y = \{yi\},\ i = 1;\ \ldots;\ |Y| \tag{6}$$

Where:

- y^i: ith target keyphrase
- $|Y|$ is the number of targets keyphrases

Thus, the sample data used for training is compatible with the target sample data, which helps to bypass problems related to partitioning in One2One model or sequencing in One2Seq model. Also, the One2Set model adopts the Transformer [44] as the encoder and decoder framework. Most of the models that generate the keyphrases are supervised. Which requires the provision of large and diverse datasets for training. Also, its performance varies according to the topics and the size of the document. Unfortunately, there is a dearth of unsupervised models. AutoKeyGen [45] is currently the only unsupervised method to generate keyphrases.

2.3 Discussion

Keyphrase prediction relies on two mechanisms: extraction and generation. Extraction approaches consist of identifying text segments already present in the source document. These segments are selected using various techniques (statistical, syntactic, or semantic). These approaches identify lexically present keyphrases in the document. However, they are not capable to generate keyphrases not mentioned in the document. They also suffer from the problem of redundancy and phrases overlap. Conversely, generation approaches allow the production of keyphrases that are not necessarily present in the source text. They are generally based on sequence-to-sequence (seq2seq) models. These approaches have the ability to produce keyphrases more consistent with human annotations, but still require large

volumes of annotated dataset for training. They also suffer from the problem of redundancy and overlap and can produce irrelevant keyphrases. Generally, there are still challenges common to both mechanisms, such as:

- Redundancy: Many models fail to filter out semantically overlapping keyphrases.
- Language sensitivity: The syntactic diversity of some languages can reduce the effectiveness of the approaches, especially in unsupervised models.
- The optimal number of keyphrases: Most methods rely on fixed thresholds or heuristics, which may not match the actual information density of the document.

In many studies [6], [46], the two mechanisms can be complementary. Extraction can serve as a first step to identify candidate phrases, while generation can reformulate or enrich these phrases.

3 KEYPHRASES PREDICTION EVALUATION

The evaluation process of keyphrases prediction helps to measure its performance for extraction or generation the most representative phrases for a document. In this section, we will introduce the different keyphrase prediction evaluation processes that were adopted. In addition, we will present the metrics and datasets that exploited these processes.

3.1 Evaluation process

According to many reviews [3], [8], [10], [47], most keyphrase prediction methods rely on performance evaluation by testing these methods on datasets that include files containing keyphrases manually identified by authors or readers. The performance of these methods is calculated by a set of metrics that compare automatically predicted keyphrases to manually selected keyphrases. Figure 6 present the evaluation process of keyphrases prediction appoaches.

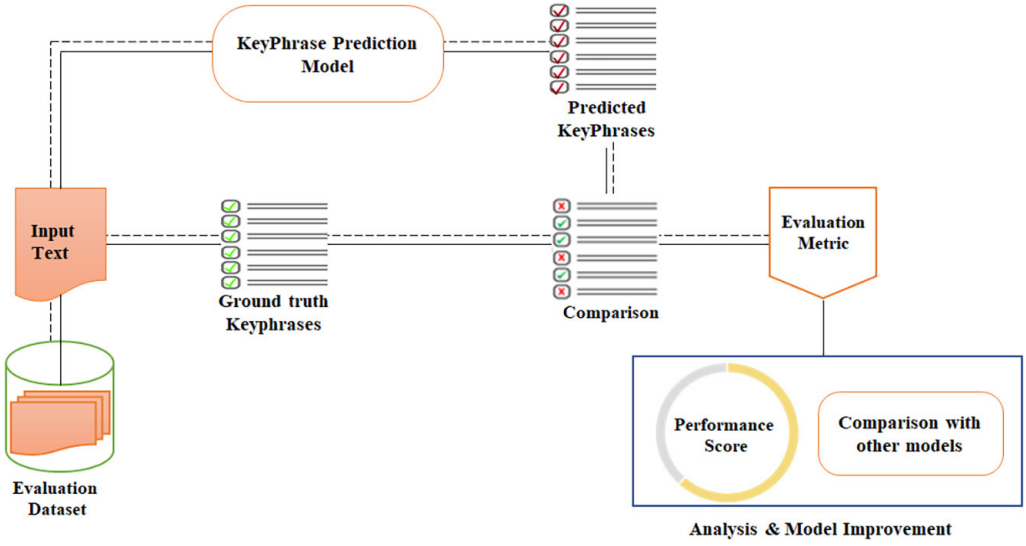

Fig. 6. Evaluation process of keyphrases prediction approaches

To analyze the evaluation process and identify their challenges, we will first study the datasets used in the evaluation. We will also study the different evaluation metrics used to calculate the keyphrase prediction performance, and the comparison of the evaluated model with other methods.

3.2 Evaluation datasets

Reference datasets selection is the first step in the evaluation process of keyphrases prediction methods. Each model tries to choose the appropriate datasets to achieve the best performance. Therefore, in Table 1 we will detail the features of different datasets in order to facilitate the selection of dataset during the evaluation process.

Table 1 shows that the datasets used to evaluate keyphrases predictions vary between academic and scientific datasets containing scientific papers such as SemEval-2010, NUS and Krapivin, or scientific abstracts papers such as Inspec, KP20k and KDD. Also, there are datasets containing diverse documents, such as DUC 2001, which contains general documents abstracts, and OpenKP, which contains annotated web pages. Also, WikiNews which contains diverse news articles. Table 1 also displays some features of each dataset, including language, number of documents, and keyphrase annotation. It also displays the average number of keyphrases and words number per document.

Table 1. Keyphrases prediction: Features of evaluation datasets

Nature	Dataset	Lang	#Docs (test)	Identification	#keyphrase	Avg Words
Scientific Papers	PubMed [48]	En	1320 (1320)	Authors	5.5	5300
	SemEval [9]	En	100 (244)	Authors/Readers	15	8000
	NUS [27]	En	211 (211)	Authors/Readers	11	8400
	Citeulike-180 [49]	En	182 (182)	Readers	5.5	8600
	ACM [50]	En	2304 (2304)	Authors	5.5	9200
	CSTR [51]	En	630 (500)	Authors	5.5	11500
Scientific Abstracts	Inspec [52]	En	500 (1500)	Indexers	10	140
	WWW [53]	En	1330 (1330)	Author	5	160
	TermITH-Eval [54]	Fr	400 (400)	Indexers	12	170
	KP20k [12]	En	20000 (547090)	Authors	5	180
	KDD [53]	En	755 (755)	Authors	4	190
	TALN-Archives [55]	En/Fr	521/1200	Authors	4	120/140
Diverse Documents	Wikinews-Kp [18]	Fr	100 (100)	Readers	9.5	320
	110-PT-BN-KP [56]	Pt	10 (110)	Readers	27.5	440
	500N-KPCrowd [57]	En	50 (500)	Readers	46	470
	DUC-2000 [58]	En	308 (308)	Readers	8	850
	KPTimes [59]	En	20000 (279923)	Editors	5.0	920

The features analysis of the datasets used to evaluate keyphrase prediction methods revealed several challenges that affect the evaluation results. They will be discussed in Section 4 by presenting the results of the review.

3.3 Evaluation metrics

To measure the performance of keyphrase prediction methods, several evaluation metrics are used [Papajeroo]. These metrics measure the accuracy and coverage of predicted keyphrases and analyze their relevance to a reference dataset. According to several studies [3], [60], the most common measures are precision, recall and F-measure. Mean Reciprocal Rank (MRR) and Average Precision (MAP) are exploited by other keyphrases prediction methods [61], [62] as evaluation metrics. There are also some works [60], [63] that have proposed measuring the semantic similarity of predicted keyphrases. In this review, we have classified evaluation metrics into three types based on the method of calculating performance:

- Lexical similarity metrics
- Ranking metrics
- Semantic similarity metrics

In this part, we will explore how to use evaluation metrics and their importance in the evaluation process according to each type.

- **Lexical similarity metrics:** Many works have used lexical similarity measures [64]. Some keyphrase prediction methods have also used them to measure their performance [2], [46], [65]. These metrics consider a predicted phrase a keyphrase if it exactly matches a manually selected keyphrase. While these metrics ensure evaluation accuracy, they ignore semantically close predictions.

  The precision metric evaluates the model's ability to reduce errors in keyphrase prediction results. The precision score is calculated using formula (7). The closer this score is to 1, this model has increased accuracy.

$$\text{Precision} = \frac{\#\text{Correctly predicted key phrases}}{\#\text{Total predicted keyphrases}} \tag{7}$$

  However, this metric remains limited in effectively evaluating the model because it does not consider unpredicted keyphrases. Therefore, it cannot be adopted without using other metrics such as recall and F-measure.

  The recall metric is used to measure a model's ability to predict the majority of the annotation keyphrases, which is important in tasks that require information retrieval. To calculate the recall value, the formula (8) is used.

$$\text{Recall} = \frac{\#\text{Correctly predicted keyphrases}}{\#\text{Total of manually annotated keyphrases}} \tag{8}$$

  However, this value is insufficient to judge the precision of the results, as generating a large number of keyphrases impacts this precision. Therefore, although the recall metric is important, it is not sufficient to fully evaluate the model.

  When the precision model is used as an evaluation metric, it may overlook some keyphrases, and when the recall model is used as an evaluation metric, it may generate many false keyphrases. To achieve balance, the F1-score is defined

as the harmonic mean between precision and recall. This metric is calculated using Formula (9).

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

The F1-Score is a useful metric for creating a balance between precision and recall, although a high F1-Score does not necessarily mean that the model is performing well.

In the keyphrases prediction task, the generated keyphrases are often not identical to the reference keyphrases. Therefore, using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric [66] enables the calculation of the partial similarity between the predicted keyphrase and the manually selected keyphrases based on the overlap of words or word groups. To calculate the ROUGE metric scores, we use the ROUGE-N (based on n-grams) variant, which is calculated using formula 10, or ROUGEL, which is calculated using formula (11).

$$ROUGE_N = \frac{\sum_{Ngrams \in reference} Count_{match}(Ngrams)}{\sum_{Ngrams \in reference} Count(Ngrams)} \tag{10}$$

Where:

- $Count_{match}$ (N grams): The number of times the common n-gram appears in the predicted and reference keyphrase.
- $Count_{match}$ (N grams): The number of occurrences of the n-gram in the reference.

To calculate the ROUGEL variant, the longest common subsequence (LCS) is used, with the same order in both phrases.

$$ROUGE_L = \frac{2 \times LCS}{P + R} \tag{11}$$

Where:

- LCS: Length of the longest common subsequence between the predicted and reference keyphrase.
- P: Length (in words) of the predicted keyphrase.
- R: Length (in words) of the reference keyphrase.

Although the ROUGE metric features the ability to calculate the partial similarity between a predicted keyphrase and annotated keyphrases based on word overlap, its adoption in keyphrase prediction methods [67], [68] is still very limited compared to other metrics.

- **Rank-based metrics:** Rank-based metrics not only evaluate the performance of models to predict keyphrases, but also measure the model's ability to rank predicted phrases according to their importance [60]. Some of the most important rank-based metrics include MRR, normalized discounted cumulative gain (NDCG), and mean average precision (MAP). These metrics helped improve prediction of top-ranking keyphrases, which is essential for natural language processing applications.

  The MAP metric [69] is one of the metrics used to evaluate the performance of keyphrase prediction models. MAP measures the model's predictions for a list of keyphrases ranked by importance. It thus assesses the model's ability to correctly classify keyphrases. MAP is calculated by averaging the average precision (AP) of

a set of documents. Formula (12) is used to calculate MAP. Formula 4 is used to calculate mean average precision.

$$MAP = \frac{1}{K}\sum_{i}^{K} Average\_Precision_i \tag{12}$$

Where:

- K: Total number of documents.
- $Average\_Precision_i$: Average precision of the i document. It is calculated using formula (13).

$$Average\_Precision_i = \frac{1}{P} \times \sum_{j=1}^{P} Precision@j \times \varphi(j) \tag{13}$$

Where:

- P: Total number of correct keyphrases
- Precision@j: Precision calculated up to position j
- φ(j): Equals 1 if the prediction at j is correct, otherwise 0.

A MAP value close to 1 means that most correct predictions are perfectly placed at the top of the list. Conversely, a low MAP indicates that most correct predictions are ranked at the bottom of the list. Correct keyphrases with lower rankings affect the MAP value and, consequently, the evaluation score. MAP is also affected by considering all correct keyphrases of equal importance. It is also affected by the size of the dataset. MAP achieves better results on larger datasets. The overlap between keyphrases also affects the MAP value. Therefore, to obtain a more comprehensive evaluation of keyphrase prediction performance, other metrics should be used in conjunction with the use of mean average precision.

The MRR metric [70] is based on the ranking of keyphrases to evaluate the performance of prediction models. MRR measures the ranking of a predicted keyphrase in a sorted list of pre-selected phrases. The higher the rank of the predicted phrase, the higher the MRR. It is measured using formula (14).

$$MRR = \frac{1}{R}\sum_{k=1}^{R} \frac{1}{rank_p(D_i)} \tag{14}$$

Where:

- R: Number of dataset documents
- $rank_p(D_i)$: Position of the first predicted keyphrase in the predictions list of the kth document

Mean Reciprocal Rank is the average of the inverse ranks of the first correct keyphrases for a set of documents. If all predicted keyphrases are not found in the list, the value of $rank_p$ is set to infinity. The MRR value is close to 1, meaning that the model predicts important keyphrases. Conversely, a low MRR value confirms the model's inability to predict important keyphrases. In contrast, MRR only considers the first predicted keyphrase, while ignoring the other keyphrases in the list. Therefore, this metric cannot judge the model's effectiveness in predicting multiple keyphrases. Therefore, MRR is often used in conjunction with other metrics to comprehensively evaluate a model.

Normalized Discounted Cumulative Gain [71], is a metric used to evaluate the quality of keyphrase prediction rankings by giving more weight to phrases that appear higher in the list of predicted keyphrases. To calculate NDCG, the Discounted Cumulative Gain (DCG) and Ideal DCG (IDCG) are first calculated using

formulas (15) and (16). DCG is calculated based on the relevance scores of the predicted keyphrases. A logarithmic function is used to reduce the importance of low-ranking phrases.

$$DCG@N = \sum_{k=1}^{N} \frac{rv_k}{\log_2(k+1)} \quad (15)$$

Where:
- N: The number of keyphrases considered
- rv_k: The relevance of the predicted keyphrase at position k

IDCG is the DCG value that would be obtained if the relevant keyphrases were perfectly ranked at the top of the list.

$$IDCG@P = \sum_{k=1}^{|H_P|} \frac{1}{\log_2(k+1)} \quad (16)$$

Where:
- $|H_P|$: number of relevant phrases among the first P results

The NDCG allows us to measure how well predicted phrases are ranked, compared to the ideal ranking. NDCG is calculated by formula (17).

$$NDCG = \frac{DCG@N}{IDCG@P} \quad (17)$$

The NDCG value ranges from 0 to 1. A value close to 1 means the model optimally ranks all relevant keyphrases at the top of the list. The model is therefore considered very effective. The closer the value is to 0.5, the more keyphrases the model predicts, but they are ranked lower in the list. As the NDCG approaches 0, this confirms that the model predicts almost no relevant keyphrases. NDCG stands out from other metrics. It not only measures how well a model predicts keyphrases, but also whether it intelligently ranks them at the top of the list.

- **Semantic similarity metrics:** BERTScore [72] and BLEURT (BERT-based Language Evaluation Understandable by Radiant Teachers) [73] are text similarity metrics. They are used to compare model-predicted keyphrases to manually annotated keyphrases to assess the precision of keyphrase prediction models.

  To evaluate the performance of a keyphrase prediction model, the BERTScore metric is based on comparing the semantic representation of the words that make up the predicted keyphrase and the reference keyphrase, using the BERT embedding technique or one of its derivatives. This is to make the evaluation result closer to human evaluation. BERTScore is calculated by the formula (18):

$$BERTScore = \frac{2}{|P| + |R|}\left(\sum_{v \in P} \max_{w \in R} sim(v,w) + \sum_{w \in R} \max_{v \in P} sim(w,v)\right) \quad (18)$$

  Where:
  - P: the word vectors of the predicted keyword phrase
  - R: the word vectors of the reference keyword phrase
  - sim(v, w): the cosine similarity between the vectors

  Although BERTScore uses semantic similarity, it faces some challenges, including the associated cost, especially with large datasets. Also, the contextual

representations of phrases may vary, as each word is aligned with the most similar word.

BLEURT is also a metric used to evaluate keyphrase prediction models. BLEURT is based on semantic similarity using a pre-trained BERT model based on manually selected reference phrases from a dataset, making it more consistent with human evaluations. Unlike BERTScore, which represents the words of a phrase independently, BLEURT encodes the phrase and represents it with a single vector. There is no mathematical formula for calculating BLEURT because it relies on deep learning model. However, it can be calculated as follows:

$$\text{BLEURT}(P, R) = \text{MLP}(\text{BERT}_{\text{CLS}}([P; R])) \tag{19}$$

Where:

- R: the reference keyphrase
- P: the predicted keyphrase
- MLP(·): A multi-layer neural network that takes the BERT encoding as input and produces a continuous score representing the semantic similarity between R and P.
- $\text{BERT}_{\text{CLS}}(\cdot)$: The function encodes the reference keyphrase R and predicted keyphrase P in BERT model, where [P; R] is the concatenation of the two phrases.

The BLEURT score is calculated by a neural network trained on manually keyphrases. It does not rely on symbolic comparisons like n-grams, but rather on deep contextual representations. The higher the BLEURT score, the closer the prediction model is to the reference in terms of content and meaning.

On the other hand, BLEURT's reliance on BERT model requires large amounts of text, which limits its ability to adapt to technical or domain-specific keyphrases. Furthermore, evaluating each pair (reference, prediction) independently limits the assessment of the diversity of the predicted keyphrase set. Moreover, it is considered one of the most expensive evaluation metrics.

4 RESULTS AND DISCUSSION

Although keyphrase prediction methods employ various NLP techniques, making them better to extract or generate keyphrases. The results of our review of the performance evaluation model for these methods show that this model cannot reliably judge the performance of keyphrase prediction methods or that the predicted phrases are semantically equivalent to the references. The review identified a set of challenges facing current evaluation model. This section will address these challenges, both related to datasets or evaluation metrics.

4.1 Dataset Challenges

An analysis of the datasets characteristics used to evaluate keyphrase prediction methods revealed several challenges that impact the results obtained during the evaluation process. In our review, we identified five major challenges when using these datasets: Data credibility, data diversity, size and coverage, linguistic complexity and rank keyphrases. The table provides a description of each of these challenges.

Table 2. Description of datasets challenges in the keyphrases prediction evaluation

Challenge	Description
Data credibility	Manual prediction of keyphrases is often influenced by subjective biases, making it difficult to judge the results of automated prediction.
Data diversity	Some datasets are restricted to a single domain. Also, a large number of datasets are in English, which limits the evaluation.
Size and coverage	Datasets are small to accurately assess the effectiveness of the model. Also, in other datasets, keyphrases are predicted for only a portion of the document, which can distort the evaluation.
Linguistic complexity	The semantic multiplicity meanings and grammatical structures in dataset texts creates complexity for prediction methods.
Rank keyphrases	Manually selected keyphrases are not ranked by importance in the dataset. Therefore, the model may predict important keyphrases, but it may not be the most accurate.

These challenges make it difficult to adopt a viable benchmark for evaluation of keyphrases prediction methods. Therefore, it is necessary to create datasets with diverse annotations tailored to the evaluation needs.

4.2 Metrics challenges

Using precision, recall, and F1-scores to evaluate keyphrase predictions poses several challenges. The most significant of these is that non-identical sentences are considered different despite their semantic similarity. Phrases that are only partially identical to a keyphrase are ignored. Evaluation results are also affected by the use of capitalization and grammatical structures. The inability to weight keyphrases according to their importance can result in a model being considered underperforming, even if it predicts important keyphrases. These challenges affect the evaluation results obtained using these metrics, especially for keyphrase generation models. Table 3 presents the evaluation results of some keyphrase prediction methods using the precision, recall, and F1-score evaluation metrics.

Overall, there are several advantages to using precision, recall, and F1-scores to evaluate keyphrase predictions, but these challenges require complementing the evaluation with more realistic and accurate metrics. The ROUGE metric, affected by the strictness of word order, receiving lower scores if the order is different. It is also biased toward phrases that are more frequent in a text. The length of phrases also affects the ROUGE metric. This metric cannot identify the most important phrases in a text.

Table 3. Evaluation results of some keyphrase prediction methods using lexical similarity metrics

Method	Inspec		NUS		Kripivin		SemEval		KP20K	
	P	A	P	A	P	A	P	A	P	A
CopyRNN [12]	0.28	0.05	0.27	0.06	0.24	0.11	0.21	0.04	0.27	0.13
CorrRNN [41]	0.25	0.04	0.26	0.06	0.24	0.11	0.22	0.04	0.26	0.11
ParaNet [42]	0.35	0.05	0.35	0.06	0.28	0.12	0.31	0.04	0.28	0.13
CatSeq [74]	0.29	0.03	0.35	0.04	0.27	0.07	0.30	0.03	0.27	0.06
GAN_{MR} [75]	0.27	0.01	0.37	0.03	0.26	0.04	0.25	–	0.25	0.03
SETTRANS [39]	0.21	0.02	0.28	0.04	0.21	0.05	0.24	0.03	0.24	0.04
WR-One2Set [76]	0.25	0.03	0.29	0.06	0.23	0.06	0.26	0.04	0.25	0.05
AutoKeyGen [77]	0.35	0.02	0.16	0.02	0.23	0.03	0.24	0.01	0.25	0.02

Notes: P: Present keyphrases; A: Absent keyphrases.

Ranking metrics only consider the first predicted keyphrases, and ignore correct keyphrases with lower rankings. Therefore, these metrics cannot evaluate the model's effectiveness for multiple keyphrases prediction. Also overlap between keyphrases also affects the evaluation result. Therefore, to obtain a more comprehensive evaluation of keyphrase prediction performance, other metrics should be used in conjunction with these metrics.

Ranking metrics only consider the first predicted keyphrases, and ignore correct keyphrases with lower rankings. Therefore, these metrics cannot evaluate the model's effectiveness for multiple keyphrases prediction. Also overlap between keyphrases also affects the evaluation result. Therefore, to obtain a more comprehensive evaluation of keyphrase prediction performance, other metrics should be used in conjunction with these metrics.

Semantic similarity metrics also have some challenges. They require large amounts of text, which limits their ability to adapt to technical or domain-specific keyphrases. Furthermore, evaluating each (reference, prediction) pair independently limits the ability to evaluate the diversity of the predicted keyphrases. Moreover, these metrics are considered one of the most expensive evaluation metrics.

The results presented in the table 4 show that not all the metrics used to evaluate keyphrases prediction models enable a comprehensive evaluation, as they all face certain challenges. Therefore, in order to accurately evaluate the performance of keyphrases prediction models, solutions must be found to overcome these challenges.

Table 4. Strengths and challenges of Keyphrases prediction evaluation metrics

Type	Metric	Strengths	Challenges
Lexical similarity	– Precision – Recall – F1-Score – ROUGE	– Easy to calculate – Clear interpretation – Lexical overlap – More useful for short keyphrases	– The semantic meaning of keyphrases – Paraphrasing – Keyphrases ranking quality
Ranking	– MAP – MRR – NDCG	– Quality of keyphrases ranking – The importance of predicted keyphrases	– Difficulty ranking keyphrases in long texts – Difficulty of interpretation – Ignoring correct keyphrases at the bottom of the list – Ignoring semantic similarity
Semantic similarity	– BERTScore – BLEURT	– Contextual Meaning Evaluation of keyphrases – Phrasing Variations – Similar to Human Evaluation	– Overlapping keyphrases – Sensitive to keyphrase length – High cost – Difficulty in interpretation

4.3 Discussion

The evaluation of keyphrase prediction models is the important step to measure their performance and improve them to ensure the quality of their results. However, this current evaluation highlights several challenges related to linguistic diversity, the predominance of human subjectivity in texts, and the keyphrases specificity in specialized texts such as biomedical documents, humanities, and computer science, etc. Therefore, the evaluation process is not just about counting the number of matching keyphrases. For the performance of the evaluated approach to be reliable, it must make an evaluation that measures the performance on three axes:

taxonomic, lexical, and semantic, which is not available through current metrics. Linguistic challenges, the size, and specificity of the texts included in the datasets used in the evaluation process must also be considered. The evaluation results are influenced by the diversity of domains and text types (tweets, abstracts, articles, etc.). Model performance is often evaluated on datasets that allow comparisons with other methods to determine which model performs best relative to others. Therefore, the choice of metrics should be thoughtful and ideally supported by human evaluations where possible to ensure a balanced and valid evaluation that will allow the model to be adopted in various NLP tasks such as information retrieval, indexation, text classification, and text summarization.

In general, when using these metrics to evaluate keyphrases prediction methods, the following aspects should be considered:

- Linguistic challenges: Linguistic complexity and domain-specific terminology can significantly impact evaluation accuracy.
- Multidimensional evaluation: Relying on a single metric is insufficient. Therefore, a multidimensional evaluation framework should be considered that considers lexical diversity and the semantic similarity of keyphrases.

These additions will help provide a more balanced and comprehensive understanding of the evaluation field and address the fundamental limitations associated with current evaluation metrics.

5 CONCLUSION

The development of techniques used in NLP field, especially deep learning models and transformer models, has contributed to the proposal of many keyphrases prediction models. In contrast, evaluation methods have not improved, making it difficult to compare the performance of these models. This paper provides a comprehensive review of the approaches to evaluating keyphrase prediction models by analyzing the different evaluation datasets. Also, the evaluation metrics used, classified into three types according to the principle adopted, and the interpretation of the results obtained. Our review found that current evaluation processes lack contextual sensitivity, especially when applied across diverse domains and languages. Therefore, to address these problems, we propose several practical directions for future research, such as developing diverse, domain-specific datasets that enable more representative benchmarks, evaluation techniques that integrate lexical-semantic metrics. Also, the future research should focus on standardizing evaluation protocols, and integrating contextual knowledge to improve the relevance of predicted keyphrases.

6 REFERENCES

[1] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in NLP," *IEEE Access*, vol. 11, pp. 36120–36146, 2023. https://doi.org/10.1109/ACCESS.2023.3266377

[2] B. Xie *et al.*, "From statistical methods to deep learning, automatic keyphrase prediction: A survey," *Information Processing & Management*, vol. 60, no. 4, p. 103382, 2023. https://doi.org/10.1016/j.ipm.2023.103382

[3] 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, 2022. https://doi.org/10.3991/ijim.v16i16.33081

[4] T. Edwin and V. Sowmya, "Keyphrase generation: Lessons from a reproducibility study," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics*, Torino, Italia, 2024, pp. 9720–9731.

[5] 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: Association for Computational Linguistics, 2023, pp. 2153–2164. https://doi.org/10.18653/v1/2023.findings-eacl.161

[6] H. Wu *et al.*, "UniKeyphrase: A unified extraction and generation framework for keyphrase prediction," *arXiv preprint arXiv:2106.04847*, 2021. https://doi.org/10.48550/ARXIV.2106.04847

[7] C. Sun, L. Hu, S. Li, T. Li, H. Li, and L. Chi, "A review of unsupervised keyphrase extraction methods using within-collection resources," *Symmetry*, vol. 12, no. 11, p. 1864, 2020. https://doi.org/10.3390/sym12111864

[8] 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*, Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 1262–1273. https://doi.org/10.3115/v1/P14-1119

[9] F. Boudin, H. Mougard, and D. Cram, "How document pre-processing affects keyphrase extraction performance," *arXiv preprint arXiv:1610.07809,* 2016. https://doi.org/10.48550/ARXIV.1610.07809

[10] E. Papagiannopoulou and G. Tsoumakas, "A review of keyphrase extraction," *arXiv preprint arXiv:1905.05044*, 2019. https://doi.org/10.48550/ARXIV.1905.05044

[11] N. Firoozeh, A. Nazarenko, F. Alizon, and B. Daille, "Keyword extraction: Issues and methods," *Nat. Lang. Eng.*, vol. 26, no. 3, pp. 259–291, 2020. https://doi.org/10.1017/S1351324919000457

[12] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 582–592. https://doi.org/10.18653/v1/P17-1054

[13] Z. Alami Merrouni, B. Frikh, and B. Ouhbi, "Automatic keyphrase extraction: A survey and trends," *J Intell. Inf. Syst.*, vol. 54, no. 2, pp. 391–424, 2020. https://doi.org/10.1007/s10844-019-00558-9

[14] O. Z. Seghroucheni, M. Lazaar, and M. Al Achhab, "Systematic review and framework for AI-driven tacit knowledge conversion methods and machine learning algorithms for ontology-based chatbots in E-learning platforms," *Int. J. Interact. Mob. Technol.*, vol. 19, no. 1, pp. 126–139, 2025. https://doi.org/10.3991/ijim.v19i01.51051

[15] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," 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: Association for Computational Linguistics, 2019, pp. 3613–3618. https://doi.org/10.18653/v1/D19-1371

[16] M.-S. Paukkeri and T. Honkela, "Likey: Unsupervised language-independent keyphrase extraction," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010, pp. 162–165.

[17] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "YAKE! Keyword extraction from single documents using multiple local features," *Information Sciences*, vol. 509, pp. 257–289, 2020. https://doi.org/10.1016/j.ins.2019.09.013

[18] A. Bougouin, F. Boudin, and B. Daille, "TopicRank: Graph-based topic ranking for keyphrase extraction," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 543–551.

[19] W. Rui, L. Wei, and M. Chris, "Corpus-independent generic key-phrase extraction using word embedding vectors," in *Software Engineering Research Conference*, 2014, pp. 1–8.

[20] C. Florescu and C. Caragea, "PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1105–1115. https://doi.org/10.18653/v1/P17-1102

[21] V. Venktesh, M. Mohania, and V. Goyal, "Topic aware contextualized embeddings for high quality phrase extraction," in *Advances in Information Retrieval*, in Lecture Notes in Computer Science, M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, and V. Setty, Eds., Cham: Springer International Publishing, vol. 13185, 2022, pp. 457–471. https://doi.org/10.1007/978-3-030-99736-6_31

[22] D. Mahata, J. Kuriakose, R. R. Shah, and R. Zimmermann, "Key2Vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 634–639. https://doi.org/10.18653/v1/N18-2100

[23] Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang, "SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model," *IEEE Access*, vol. 8, pp. 10896–10906, 2020. https://doi.org/10.1109/ACCESS.2020.2965087

[24] K. Patel and C. Caragea, "Exploiting position and contextual word embeddings for keyphrase extraction from scientific papers," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online: Association for Computational Linguistics, 2021, pp. 1585–1591. https://doi.org/10.18653/v1/2021.eacl-main.136

[25] P. D. Turney, "Learning algorithms for keyphrase extraction," *Information Retrieval*, vol. 2, no. 4, pp. 303–336, 2000. https://doi.org/10.1023/A:1009976227802

[26] P. D. Turney, "Coherent keyphrase extraction via web mining," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03),* Acapulco, Mexico, 2003, pp. 434–439. https://doi.org/10.48550/ARXIV.CS/0308033

[27] T. D. Nguyen and M.-Y. Kan, "Keyphrase extraction in scientific publications," in *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, in Lecture Notes in Computer Science, D. H.-L. Goh, T. H. Cao, I. T. Sølvberg, and E. Rasmussen, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 4822, 2007, pp. 317–326. https://doi.org/10.1007/978-3-540-77094-7_41

[28] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, and B. Wang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computer Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.

[29] K. Zhang, H. Xu, J. Tang, and J. Li, "Keyword extraction using support vector machine," in *Advances in Web-Age Information Management*, in Lecture Notes in Computer Science, J. X. Yu, M. Kitsuregawa, and H. V. Leong, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 4016, 2006, pp. 85–96. https://doi.org/10.1007/11775300_8

[30] X. Jiang, Y. Hu, and H. Li, "A ranking approach to keyphrase extraction," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, USA: ACM, 2009, pp. 756–757. https://doi.org/10.1145/1571941.1572113

[31] K. Sarkar, M. Nasipuri, and S. Ghose, "A new approach to keyphrase extraction using neural networks," *arXiv preprint arXiv:1004.3274,* 2010. https://doi.org/10.48550/ARXIV.1004.3274

[32] J. Villmow, M. Wrzalik, D. Krechel, and P. Perner, Eds., "Automatic keyphrase extraction using recurrent neural networks," in *Machine Learning and Data Mining in Pattern Recognition*, in Lecture Notes in Computer Science, Cham: Springer International Publishing, vol. 10935, 2018, pp. 210–217. https://doi.org/10.1007/978-3-319-96133-0_16

[33] Y. Luo, Y. Xu, J. Ye, X. Qiu, and Q. Zhang, "Keyphrase generation with fine-grained evaluation-guided reinforcement learning," *arXiv preprint arXiv:2104.08799*, 2021. https://doi.org/10.48550/ARXIV.2104.08799

[34] D. Wu, W. U. Ahmad, and K.-W. Chang, "Pre-trained language models for keyphrase generation: A thorough empirical study," *arXiv preprint arXiv:2212.10233*, 2022. https://doi.org/10.48550/ARXIV.2212.10233

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014. https://doi.org/10.48550/ARXIV.1409.3215

[36] W. Chen, H. P. Chan, P. Li, L. Bing, and I. King, "An integrated approach for keyphrase generation via exploring the power of retrieval and extraction," *arXiv preprint arXiv:1904.03454*, 2019. https://doi.org/10.48550/ARXIV.1904.03454

[37] F. Boudin and Y. Gallina, "Redefining absent keyphrases and their effect on retrieval effectiveness," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, 2021, pp. 4185–4193. https://doi.org/10.18653/v1/2021.naacl-main.330

[38] X. Yuan *et al.*, "One size does not fit all: Generating and evaluating variable number of keyphrases," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 7961–7975. https://doi.org/10.18653/v1/2020.acl-main.710

[39] J. Ye, T. Gui, Y. Luo, Y. Xu, and Q. Zhang, "One2Set: Generating diverse keyphrases as a set," *arXiv preprint arXiv:2105.11134*, 2021. https://doi.org/10.48550/ARXIV.2105.11134

[40] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. https://doi.org/10.48550/ARXIV.1409.0473

[41] J. Chen, X. Zhang, Y. Wu, Z. Yan, and Z. Li, "Keyphrase generation with correlation constraints," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4057–4066. https://doi.org/10.18653/v1/D18-1439

[42] J. Zhao and Y. Zhang, "Incorporating linguistic constraints into keyphrase generation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 5224–5233. https://doi.org/10.18653/v1/P19-1515

[43] R. Meng, X. Yuan, T. Wang, S. Zhao, A. Trischler, and D. He, "An empirical study on neural keyphrase generation," *arXiv preprint arXiv:2009.10229*, 2020. https://doi.org/10.48550/ARXIV.2009.10229

[44] A. Vaswani *et al.*, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017. https://doi.org/10.48550/ARXIV.1706.03762

[45] X. Shen, Y. Wang, R. Meng, and J. Shang, "Unsupervised deep keyphrase generation," *arXiv preprint arXiv:2104.08729*, 2021. https://doi.org/10.48550/ARXIV.2104.08729

[46] L. Ajallouda, F. Z. Fagroud, A. Zellou, and E. B. Lahmar, "KP-USE: An unsupervised approach for key-phrases extraction from documents," *IJACSA*, vol. 13, no. 4, 2022. https://doi.org/10.14569/IJACSA.2022.0130433

[47] Z. Nasar, S. W. Jaffry, and M. K. Malik, "Textual keyword extraction and summarization: State-of-the-art," *Information Processing & Management*, vol. 56, no. 6, p. 102088, 2019. https://doi.org/10.1016/j.ipm.2019.102088

[48] A. Thorsten Schutz, "Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical," Master's thesis, National University of Ireland, 2008.

[49] O. Medelyan, E. Frank, and I. H. Witten, "Human-competitive tagging using automatic keyphrase extraction," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09)*, Singapore: Association for Computational Linguistics, 2009, p. 1318. https://doi.org/10.3115/1699648.1699678

[50] K. Mikalai, A. Aliaksandr, and M. Maurizio, "Large dataset for keyphrases extraction," University of Trento, 2009.

[51] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," *arXiv preprint arXiv:9902007*, 1999. https://doi.org/10.48550/ARXIV.CS/9902007

[52] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2003, pp. 216–223. https://doi.org/10.3115/1119355.1119383

[53] C. Caragea, F. A. Bulgarov, A. Godea, and S. Das Gollapalli, "Citation-enhanced keyphrase extraction from research papers: A supervised approach," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1435–1446. https://doi.org/10.3115/v1/D14-1150

[54] A. Bougouin, S. Barreaux, L. Romary, F. Boudin, and B. Daille, "TermITH-Eval: A French standard-based resource for keyphrase extraction evaluation," in *Language Resources and Evaluation Conference (LREC)*, 2016.

[55] F. Boudin, "TALN Archives: A digital archive of French research articles in natural language processing," in *TALN-RÉCITAL 2013, 17–21 Juin, Les Sables d'Olonne*, 2013.

[56] L. Marujo, M. Viveiros, and J. P. da S. Neto, "Keyphrase cloud generation of broadcast news," *arXiv preprint arXiv:1306.4606*, 2013. https://doi.org/10.48550/ARXIV.1306.4606

[57] L. Marujo, A. Gershman, J. Carbonell, R. Frederking, and J. P. Neto, "Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization," *arXiv preprint arXiv:1306.4886*, 2013. https://doi.org/10.48550/ARXIV.1306.4886

[58] X. Wan and J. Xiao, "Single document keyphrase extraction using neighborhood knowledge," in *AAAI '08: Proceedings of the 23rd National Conference on Artificial Intelligence*, vol. 2, 2008, pp. 855–860.

[59] Y. Gallina, F. Boudin, and B. Daille, "KPTimes: A large-scale dataset for keyphrase generation on news documents," *arXiv preprint arXiv:1911.12559*, 2019. https://doi.org/10.48550/ARXIV.1911.12559

[60] D. Wu, D. Yin, and K.-W. Chang, "KPEval: Towards fine-grained semantic-based keyphrase evaluation," *arXiv preprint arXiv:2303.15422*, 2023. https://doi.org/10.48550/ARXIV.2303.15422

[61] Y. Gallina, F. Boudin, and B. Daille, "Large-scale evaluation of keyphrase extraction models," *arXiv preprint arXiv:2003.04628*, 2020. https://doi.org/10.48550/ARXIV.2003.04628

[62] M. Nadim, D. Akopian, and A. Matamoros, "A comparative assessment of unsupervised keyword extraction tools," *IEEE Access*, vol. 11, pp. 144778–144798, 2023. https://doi.org/10.1109/ACCESS.2023.3344032

[63] 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, 2024. https://doi.org/10.1007/s10115-024-02164-w

[64] M. H. Hoti, F. Qorrolli, and F. Spahija, "Enhancing fake news detection via stance analysis: Leveraging advanced NLP techniques and machine learning models," *Int. J. Interact. Mob. Technol.*, vol. 19, no. 11, pp. 39–50, 2025. https://doi.org/10.3991/ijim.v19i11.55007

[65] T. Schopf, S. Klimek, and F. Matthes, "PatternRank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction," in *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Valletta, Malta: SCITEPRESS – Science and Technology Publications, 2022, pp. 243–248. https://doi.org/10.5220/0011546600003335

[66] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, pp. 74–81, 2004.

[67] A. V. Glazkova and D. A. Morozov, "Applying transformer-based text summarization for keyphrase generation," *Lobachevskii J Math*, vol. 44, no. 1, pp. 123–136, 2023. https://doi.org/10.1134/S1995080223010134

[68] S. Takeshita, S. P. Ponzetto, and K. Eckert, "ROUGE-K: Do your summaries have keywords?" *arXiv preprint arXiv:2403.05186*, 2024. https://doi.org/10.48550/ARXIV.2403.05186

[69] L. Liu and M. T. Özsu, Eds., *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009. https://doi.org/10.1007/978-0-387-39940-9

[70] E. M. Voorhees, "The TREC question answering track," *Nat. Lang. Eng.*, vol. 7, no. 4, pp. 361–378, 2001. https://doi.org/10.1017/S1351324901002789

[71] Y. Wang, L. Wang, Y. Li, D. He, T.-Y. Liu, and W. Chen, "A theoretical analysis of NDCG type ranking measures," *arXiv preprint arXiv:1304.6480*, 2013. https://doi.org/10.48550/ARXIV.1304.6480

[72] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," *arXiv preprint arXiv:1904.09675*, 2019. https://doi.org/10.48550/ARXIV.1904.09675

[73] T. Sellam, D. Das, and A. P. Parikh, "BLEURT: Learning robust metrics for text generation," *arXiv preprint arXiv:2004.04696*, 2020. https://doi.org/10.48550/ARXIV.2004.04696

[74] H. P. Chan, W. Chen, L. Wang, and I. King, "Neural keyphrase generation via reinforcement learning with adaptive rewards," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 2163–2174. https://doi.org/10.18653/v1/P19-1208

[75] A. Swaminathan, H. Zhang, D. Mahata, R. Gosangi, R. R. Shah, and A. Stent, "A preliminary exploration of GANs for keyphrase generation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, 2020, pp. 8021–8030. https://doi.org/10.18653/v1/2020.emnlp-main.645

[76] B. Xie *et al.*, "WR-ONE2SET: Towards well-calibrated keyphrase generation," *arXiv preprint arXiv:2211.06862*, 2022. https://doi.org/10.48550/ARXIV.2211.06862

[77] X. Shen, Y. Wang, R. Meng, and J. Shang, "Unsupervised deep keyphrase generation," *arXiv preprint arXiv:2104.08729*, 2021. https://doi.org/10.48550/ARXIV.2104.08729

7 AUTHORS

Amine Berquedich, is a PhD Student at Computer Science and Systems Analysis School (ENSIAS), SPM-ENSIAS, Mohamed V University, Rabat, Morocco. His research interests are primarily in the area of Natural language processing, search engines, deep learning, and machine learning, where he is the author/co-author of over 5 research publications (E-mail: amine_berquedich@um5.ac.ma).

Lahbib Ajallouda received his Ph.D. in Computer Science from the Computer Science and Systems Analysis School (ENSIAS), SPM-ENSIAS, Mohamed V University, Rabat, Morocco. He is currently a Professor in Computer Science at Polydisciplinary Faculty of Es-Semara, Ibn Zohr University in Agadir, Morocco. His research interests are primarily in the area of Natural language processing, Internet of Things, search

engines, cloud computing, and machine learning, where he is the author/co-author of over 17 research publications (E-mail: l.ajallouda@uiz.ac.ma).

Ahmed Zellou received his Ph.D. in Applied Sciences from the Mohammedia School of Engineers, Mohammed V University, Rabat, Morocco 2008. He is currently a coordinator of the IWIM Web Engineering & Mobile Computing branch at SPM-ENSIAS, Mohamed V university in Rabat, Morocco. His research interests include Parallel Computing, Information Systems (Business Informatics), and Distributed Computing, where he is the author/co-author of over 120 research publications (E-mail: ahmed.zellou@um5.ac.ma).

Inès Chihi is the Professor in Electrical Engineering/Head of the research laboratory Advanced Engineering and Smart Sensors Solutions (Ae3S) at Department of Engineering, Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg City, Luxembourg. His research interests include modeling of complex systems, control and fault detection, of complex systems with unpredictable behaviors. She is skilled in Electrical Engineering, Artificial Intelligence, Machine Learning, Statistics and Data Analysis. She is the author/co-author of over 60 research publications (E-mail: ines.chihi@uni.lu).