

# Present and absent keyphrases extraction: an approach based on sentence embedding

Lahbib Ajallouda, Ahmed Zellou

Software Project Management Research Team, Web and Mobile Engineering Department, ENSIAS, Mohammed V University, Rabat, Morocco

## Article Info

### Article history:

Received Mar 4, 2022

Revised Jul 26, 2022

Accepted Aug 29, 2022

### Keywords:

Automatic keyphrase extraction

Generate absent keyphrases

Natural language processing

Sentence embedding technique

Universal sentence encoder

## ABSTRACT

The automatic keyphrases extraction (AKE) of a document is any expression by which we can learn its content without having to read it. Keyphrases are exploited in natural language processing (NLP) applications. These phrases are often mentioned in the document but there may be some keyphrases that are not mentioned. In the field of AKE, researchers have exploited many techniques, such as statistical calculation, deep learning algorithms, graph representation, and sentence embedding techniques. Approaches that exploit embedding techniques calculate the similarity between a document and a candidate keyphrase, where similar phrases to the document are considered as keyphrases. Representing the document by a single vector makes its performance poor, especially in long documents. This is in addition to the inability of these methods to generate absent keyphrases. In order to overcome these problems, our paper proposes an unsupervised approach to AKE, based on the universal sentence encoder (USE) to represent candidate keyphrases and parts of the document probably containing keyphrases. Our method also generates keyphrases not mentioned in the text. We compared the performance of the proposed approach with other methods based on embedding techniques, where the results showed the superiority of our approach especially in long documents.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Lahbib Ajallouda

Software Project Management Research Team, ENSIAS Mohammed V University

Rabat, Morocco

Email: lahbib\_ajallouda@um5.ac.ma

## 1. INTRODUCTION

The quantitative production of digital documents forced the search for solutions that could summarize or analyze their content without having to read them [1]. Automatic keyphrase extraction (AKE) remains the best way to solve this problem. According to Papagiannopoulou and Tsoumakas in [2] four techniques were exploited in unsupervised AKE methods, namely statistics-based, graph-based ranking, language model-based and sentence embeddings. The sentence embedding technique [3] is one of the recent methods used by researchers to represent the document and the candidate phrases in order to measure the semantic similarity between them and to consider the phrases closest to the document as keyphrases. The performance of most AKE methods using embedding techniques remains low, especially in long documents, which contain information that is not relevant to the document's topics, so the vector representing the document does not reflect its content, making the semantic similarity value between the document and candidate keyphrases inaccurate. Keyphrases often express the document's title, abstract, or conclusions. So, this factor must be taken into account when measuring the similarity between the document and the candidate keyphrases.

The objective of this paper is to propose an unsupervised AKE method based on a technique of sentence embedding, which takes into account the localization of the keyphrases in the document during the process of measuring the similarity of the candidate keyphrases with the content of document. According to Ajalloua *et al.* [4], the best technique that can be used to represent noun phrases is universal sentence encoder (USE) [5]. Since candidate keyphrases are noun phrases, we have chosen this technique to represent candidate phrases and parts of the document that may include keyphrases. In order to calculate the more precise semantic similarity, we used weighting coefficients for these parts, taking into account the proximity of the candidate phrases to these parts. The proposed approach not only extracts keyphrases contained in the document, but also generates keyphrases which are not mentioned in the document.

We have organized the paper as follows. Section 2 presents related works. Section 3 is dedicated to presenting the proposed method for extracting and generating keyphrases from a document. Section 4 includes the results and discussion of the performance of the proposed approach in extracting the present keyphrases and generating the absent keyphrases. Section 5 includes conclusions and future directions for research.

## 2. RELATED WORKS

This section will allow us to present relevant work in the proposed method. We will first introduce the most common sentence embedding techniques and mention the advantages of each technique. We will then introduce the most important AKE methods, especially those based on embedding techniques. Next, we will present the methods that generate absent keyphrases.

### 2.1. Sentence embedding techniques

Text data is more complex than other types of data, which makes it difficult for the machine to handle. Sentence embedding technique is the ideal way to convert text of different lengths into vectors of the same dimension. Most of the studies that were concerned with text embedding methods classified these methods into: Methods that employ the technique of calculating the average vectors of the words constituting the sentence, such as smooth inverse frequency (SIF) [6] and geometric embedding (GEM) [7] where this resultant vector is considered representative of the sentence. Although this method is simple, the sentence loses its semantics due to neglect of word order. This makes sentence embedding less semantically accurate. Thanks to the advent of encryption software and deep learning algorithms, this problem has been overcome. InferSent [8] is a recurrent neural network (RNN) based embedding technique that predicts semantic relationships between sentences. Universal sentence encoder [5] is a transformer-based method for embedding text, and it can also be employed by the deep average network (DAN) that gives better results in short texts. Sentence bidirectional encoder representations from transformers (SBERT) [9], this technique employs BERT [10] in addition to a Siamese grid for string embedding. In general, although these techniques give excellent results, compared to traditional methods, their disadvantage is that they are computationally expensive and require a great effort for training. Keyphrases are rare in some documents such as biomedical documents [11]–[13], which is reflected in the performance of methods using embedding techniques. Therefore, it is necessary to generate keyphrases instead of extracting them into biomedical documents.

### 2.2. Keyphrases extraction approaches

Several AKE methods have been published in recent years. Siddiqi and Sharan [14], there are supervised methods that treat the problem of keyphrase extraction as binary classification, and unsupervised methods that rely on extracting keyphrases based on their order. There are also semi-supervised methods, which are a combination of the two previous approaches. Some studies have categorized these methods by the type of technique used to extract keyphrases. Papagiannopoulou and Tsoumakas [2], there are approaches that exploit binary classification algorithms such as [15], [16]. Most of these methods are supervised. The statistical model has also been exploited in some methods such as [17]–[19]. Most of them are unsupervised. The most famous AKE methods are those that rely on graph techniques such as [20]–[22]. One of the recent methods that has achieved the best performance is the methods that use deep learning algorithms such as [23]–[25]. The development of sentence embedding techniques also contributed to the emergence of AKE methods that use these techniques as [26]–[28].

### 2.3. Keyphrase generation approaches

In some documents, some key phrases may not be mentioned in the document. Therefore, it is not enough to extract only the keyphrases mentioned. For this, some AKE methods generate keyphrases that are not mentioned in the document. Crawshaw [29], the authors propose to generate keyphrases using an encoder that predicts the semantic meaning of a phrase through the recurrent RNN algorithm. This method created a

few problems, the biggest of which was creating phrases with the same meaning. To overcome this problem, the authors of [30] suggested using correlational recurrent neural networks (CorrRNN), to generate keyphrases covering the topics of the document. The downside of this method is the huge amount of data used for training. To reduce this amount, the authors of [31] suggested creating a topic-based adversarial neural network (TANN) that uses both labeled and unlabeled data to reduce the amount of data used in training. Pang and Lee [32] also suggested exploiting the following and preceding context of a sentence to predict key phrases, using the bidirectional long short term memory (Bi-LSTM) RNN. Rabby *et al.* [33] proposed a model using the seq2seq RNN, which extracts existing keyphrases and predicts the not mentioned in the document by exploiting linguistic, semantic and statistical information. Nguyen and Kan [34] also suggested a method that uses the keyphrases mentioned in the text, in order to construct keyphrases not mentioned in the text using the mask-predict method.

### 3. KEYPHRASE EXTRACTION AND GENERATION METHOD

#### 3.1. Process of the proposed method

The process of our method proposed in this article consists of six main steps, as shown in Figure 1. Where in the first step we extract the candidate keyphrases, to achieve this we adopted the approach proposed in [35]. The second step is dedicated to embedding the candidate keyphrases by the transformer model, while the paragraphs of the document will be embedded by the deep average network (DAN) model. The third step is to identify parts of the document that may contain keyphrases. The fourth step will be devoted to measuring the similarity between the candidate key phrases and the paragraphs of the document, as well as the extraction of the keyphrases mentioned in the text, while the fifth step will be devoted to the generation of keyphrases not mentioned in the document. We will conclude the process by deleting phrases with the same semantic meaning.

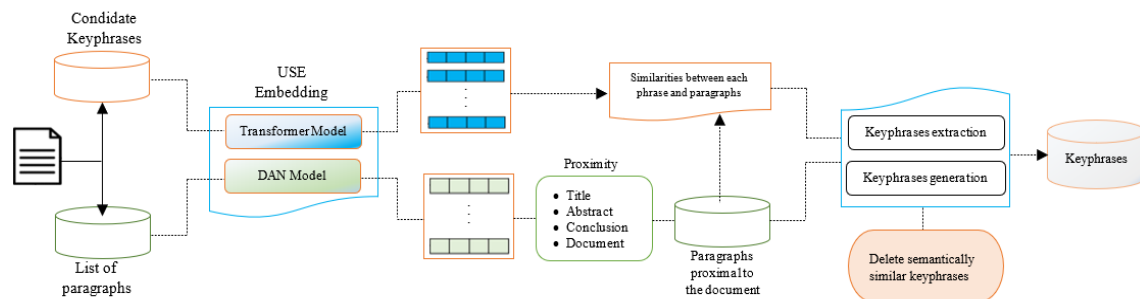


Figure 1. Present and absent keyphrases extraction process

#### 3.2. Candidate keyphrases

One of the challenges faced by AKE methods is identifying candidate keyphrases in a document. For this, many techniques have been used such as term frequency-inverse document frequency (TFIDF), N-Gram, and part-of-speech tagging (POST). The method which has been proposed in [35] gives acceptable results. Figure 2 presents the different steps in order to select candidate keyphrases. Our method adopted the same process for identifying candidate key phrases. This made us get rid of many unimportant phrases in the document.

#### 3.3. Sentence and paragraph embedding

The keyphrase extraction method that we propose in this paper is based on the similarity calculation between the candidate keyphrases and the paragraphs that will be selected in the third step. The calculation of the semantic similarity between two texts requires their vectorial representation. Several sentence embedding techniques can be used to represent paragraphs, and candidate keyphrases. Ajallouda *et al.* [4], noun phrases are best represented using the universal sentence encoder (USE) technique. We chose to use this technique because the candidate keyphrases are noun phrases.

The universal sentence encoder is a recent technique used to represent a sentence or a paragraph by a vector of 512 dimensions. First, USE encodes the phrases using an encoder that converts these phrases into vectors, which are used in some NLP tasks. The incorrect results obtained from these tasks are exploited in order to improve the vector representation of the phrases. Figure 3 shows USE process.

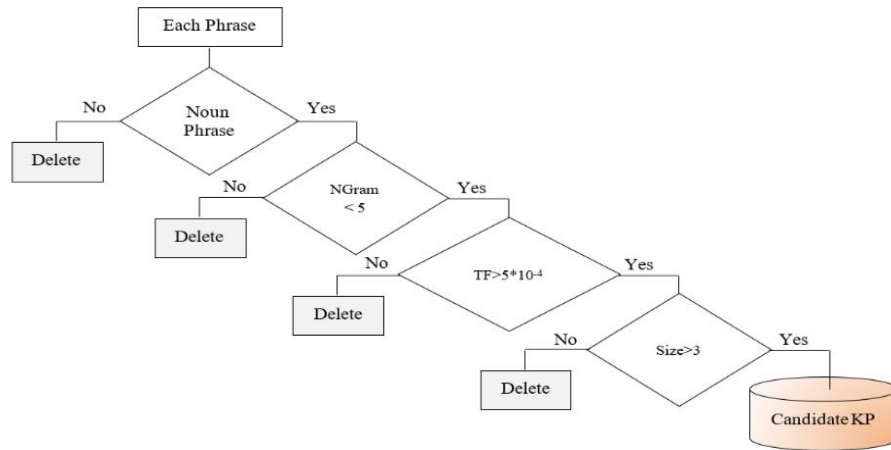


Figure 2. Candidate keyphrase selection process

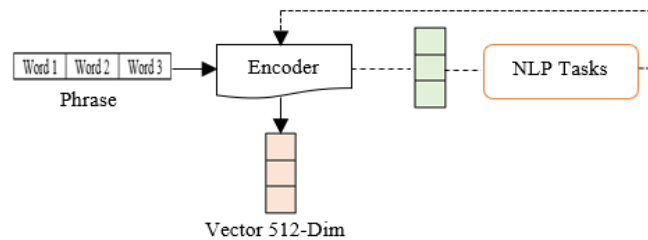


Figure 3. Universal sentence encoder process

The encoder process can be done in USE either by a transformer of 6 layers, each one containing a self-attention model and a feed forward network that enables the transformer to exploit the context and the word order during the embedding process. Encoding can also be accomplished via the deep average network (DAN) [36]. This is done by generating the average vectors of the words that make up the phrase. This vector passes by a four-layers deep neural network (DNN) that produces a vector with 512 dimensions. Figure 4 presents the encoder exploited by USE.

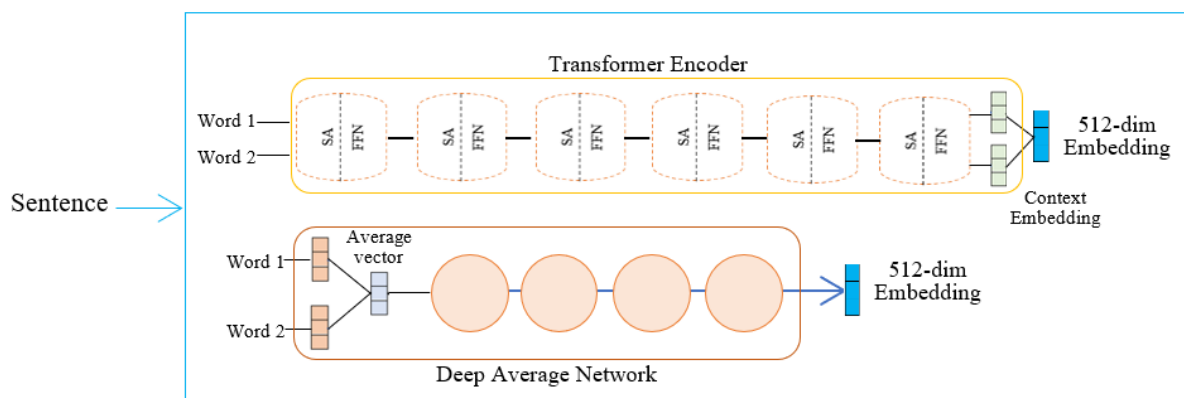


Figure 4. Transformer encoder and deep average network process

Empirical results conducted on USE in 6 datasets (See Table 1) confirmed that transformer encoding performs better than DAN encoding. On the other hand, it is difficult to use the transformer to encrypt long texts because it requires more time than DAN.

Table 1. USE performance via transform model and DAN model

Dataset	Transformer model	DAN model	Task
Customer reviews [31]	87.43	80.97	Product reviews
Movie reviews [37]	81.44	74.45	Sentiment
Multi-perspective question and answering [38]	86.98	85.38	Opinion polarity
Stanford sentiment analysis [39]	85.38	77.62	Sentiment
Subjectivity summarization [32]	93.87	92.65	Subjectivity/Objectivity
Text retrieval conference [32]	92.51	91.19	Question and answering

Which is pointed out by the USE authors who consider the complexity of transformer model to be  $O(n^2)$ , while DAN model is  $O(n)$ . For this we have chosen in our method to use the model based on the transformer to encode the candidate keyphrases while the encoding of the paragraphs will be done via the model based on DAN.

### 3.4. Score paragraph

The purpose of this step is to identify the paragraphs that express the content of the document and to get rid of irrelevant paragraphs. To identify the paragraphs expressing the document, we started from the assumption that these paragraphs are semantically similar to most of the paragraphs of the document, in particular title, abstract and conclusion. For this, we first calculate the score of each paragraph using (1).

$$Score(P_i) = \frac{1}{3} [2 \times (Sim(V_i, T) + Sim(V_i, A) + Sim(V_i, C)) + \frac{1}{N} \sum_{j=1}^N Sim(V_i, V_j)] \tag{1}$$

N: number of paragraphs in document

$V_i$ : vector represents paragraph i

$V_j$ : vector represents paragraph j

T: vector represents the title

A: vector represents the abstract

C: vector represents the conclusion

Recently, deep learning methods that calculate semantic similarity between texts have appeared [40]. The disadvantage of these methods is that their complexity is high, as well as the need to provide data for training. For this, we have chosen cosine as a measure of semantic similarity between two texts. Cosine measure is calculated using (2).

$$Sim(U, V) = \frac{U \cdot V}{\|U\| \|V\|} \tag{2}$$

$$S = \frac{Max(Score) + Min(Score)}{2} \tag{3}$$

We have defined the threshold for selecting paragraphs similar to the document in (3). Each paragraph with a score greater than or equal to S, will be considered similar to the document. This formula gave us better results than using all the paragraphs or adopting the scores average as a selection threshold.

### 3.5. Keyphrase extraction

In the previous paragraph, the paragraphs expressing the document were identified. Therefore, the candidate key phrases that are most semantically similar to these paragraphs are the keyphrases. In order to calculate the proximity of each candidate keyphrase to the paragraphs representing the document, we used (4).

$$SKP(CP_i) = \frac{\sum_{j=1}^M Score(P_j) \times Sim(U_i, V_j)}{\sum_{k=1}^M Score(P_k)} \tag{4}$$

$CP_i$ : candidate keyphrase i.

$Score(P_j)$ : score of paragraph  $P_j$ , calculated via formula 1.

$Sim(U_i, V_j)$ : similarity between  $U_i$ , vector of candidate keyphrase  $CP_i$  and  $V_j$ , vector of paragraph  $P_j$ .

M: the number of paragraphs expressing the document.

Candidate keyphrases will be ranked according to results of (4). The phrases with the highest score will be considered present keyphrases. We experimentally determined the number of appropriate keyphrases. These experiments will be presented in the results section. The selected keyphrases will be used in the absent keyphrase prediction process.

### 3.6. Keyphrase generation

When reading and analyzing a document, we often find that there are sentences that can clearly express its content even though they are not mentioned in the document. These phrases are called absent keyphrases. Also, analysis of datasets used in training and evaluation of AKE methods confirms this, as we find that almost half of the keyphrases in the document are absent keyphrases. Table 2 shows the percentage of absent key phrases in the most popular datasets.

Table 2. The percentage of present and absent keyphrases in some datasets

Dataset	Type	Docs	Present KP (%)	Absent KP (%)
SemEval	Papers	244	42.60	57.40
NUS	Papers	211	54.40	45.60
Krapivin	Papers	2 304	55.70	44.30
KPTimes	News	259 923	58.80	41.20
KP20K	Abstracts	527 090	62.90	37.10
Inspec	Abstracts	1000	73.60	26.40

This fact forced us to improve our approach to make it able to predict the absent keyphrases. For this we used the RNN encoder-decoder model of [41], [42]. This model (also called Seq2Seq) is used by deep learning methods that predict keyphrases [23], [43], [44]. However, most of these methods generate absent keyphrases but some do not express the content of the document. Our method mitigates this problem by replacing the source document in the form with the main paragraphs, which were identified in the third step. This is in addition to exploiting the previously defined keyphrases. This eliminated irrelevant texts and reduced the amount of model training data.

### 3.7. Keyphrases selection

To avoid selecting duplicate keyphrases, phrases that are part of other phrases, or irrelevant phrases. After the keyphrases extraction and generation process is completed, our method by Algorithm 1 filters all these phrases to remove duplicate keyphrases, that are part of other phrases, for example a more expressive computer sentence than computer or science. The algorithm also removes irrelevant phrases that do not express the document.

#### Algorithm 1. Keyphrases selection

```

Input: EKP, list of extracted keyphrases
         GKP, list of generated keyphrases
Output: List of selected keyphrases
Begin
  KP ← []
  // Remove duplicate keyphrases and parts keyphrases
  for i=0 to len (GKP) do
    for j=0 to len (EKP) do
      if (duplicate(GKP[i], EKP[j]))
        remove(GKP[i])
        i=i-1
      else
        if (part(GKP[i], EKP[j])) // GKP[i] is part of EKP[j] or vice versa
          if (score(GKP[i]) < score(EKP[j]))
            remove(GKP[i])
            i=i-1
          else
            remove(EKP[j])
            j=j-1
          end if
        end if
      end if
    end for
  end for
  // Remove irrelevant keyphrases
  EKP.append(GKP)
  KP ← []
  for i=1 to len(EKP) do
    if (score(EKP[i]) >=(scoreMax+scoreMin)/2)
      KP.append(EKP[i])
    end if
  end for
  return KP
end

```

This algorithm allowed us to select the list of keyphrases present or absent, the most expressive of the document. Our method will rank them according to their scores calculated using (4). The phrases that rank first are the keyphrases of the document.

#### 4. RESULTS AND DISCUSSION

In the first part of the results and discussion section, we will present the tools that have been exploited to evaluate our method. We will first describe the data sets used in training and testing, as well as the evaluation metrics used. We will then introduce a method for selecting the paragraphs that express the document. In the second part, we will discuss the results obtained and compare them with the performance of other AKE methods.

##### 4.1. Datasets

To evaluate AKE methods. The available datasets can be used. However, recurrent neural network training requires a dataset that has a huge number of documents. This is provided by the KP20k [23] dataset, which contains 527,830 articles, of which 40,000 posts are randomly selected, with 20,000 articles devoted to training while the rest are used for testing. That's why we chose this dataset to train the part of generating absent keyphrases.

In addition to KP20k we will evaluate our method on the Inspec [45] dataset, containing abstract scientific papers which will enable us to measure performance in short texts. We will also use the Semeval2010 [46] and NUS [34] datasets, containing scientific papers to evaluate the performance of our method in long texts.

##### 4.2. Evaluation metrics

Although several metrics are available to evaluate the performance of AKE methods [47]. However, most researchers prefer to use only three measures, recall (5) which expresses the number of keyphrases extracted from among the keyphrases of the document. Precision (6) which expresses the number of valid keyphrases extracted of the total keyphrases extracted. F1-measure (7) which is calculated to express interaction and to combine precision and recall.

$$Recall = \frac{True\ Keyphrases}{KeyPhrases\ extracted\ by\ the\ author} \quad (5)$$

$$Precision = \frac{True\ KeyPhrases}{True\ KeyPhrases+False\ KeyPhrases} \quad (6)$$

$$F1.\ Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

Its ease of use and the precision of its results are what made most researchers prefer to evaluate AKE methods using these metrics. We will use these metrics in order to evaluate and compare the performance of our method in other ways.

##### 4.3. Paragraphs of the document

Before the process of extracting the present keyphrases, the paragraphs that express the content of the document are selected, by calculating the score of each paragraph by (1). The paragraphs with the highest score are chosen as paragraphs expressing the document. To select the appropriate number of paragraphs, we tried three formulas, the first case is to keep all the paragraphs, the second case is to use formula 3 while the third case is to choose the paragraphs that have a score greater than the average score of all paragraphs, which is calculated by (8).

$$AvgScore = \frac{1}{N} \sum_{i=1}^N Score(P_i) \quad (8)$$

N: number of paragraphs in document

Score (P<sub>i</sub>): the score of paragraph i, calculated via (1).

We applied these three cases to all datasets, where the results obtained (see Figure 5) showed that the second case outperforms the other cases in all datasets, which determines the appropriate number of

closest paragraphs of the document using (3). Which is the same number of paragraphs we used in the process of generating the absent keyphrases.

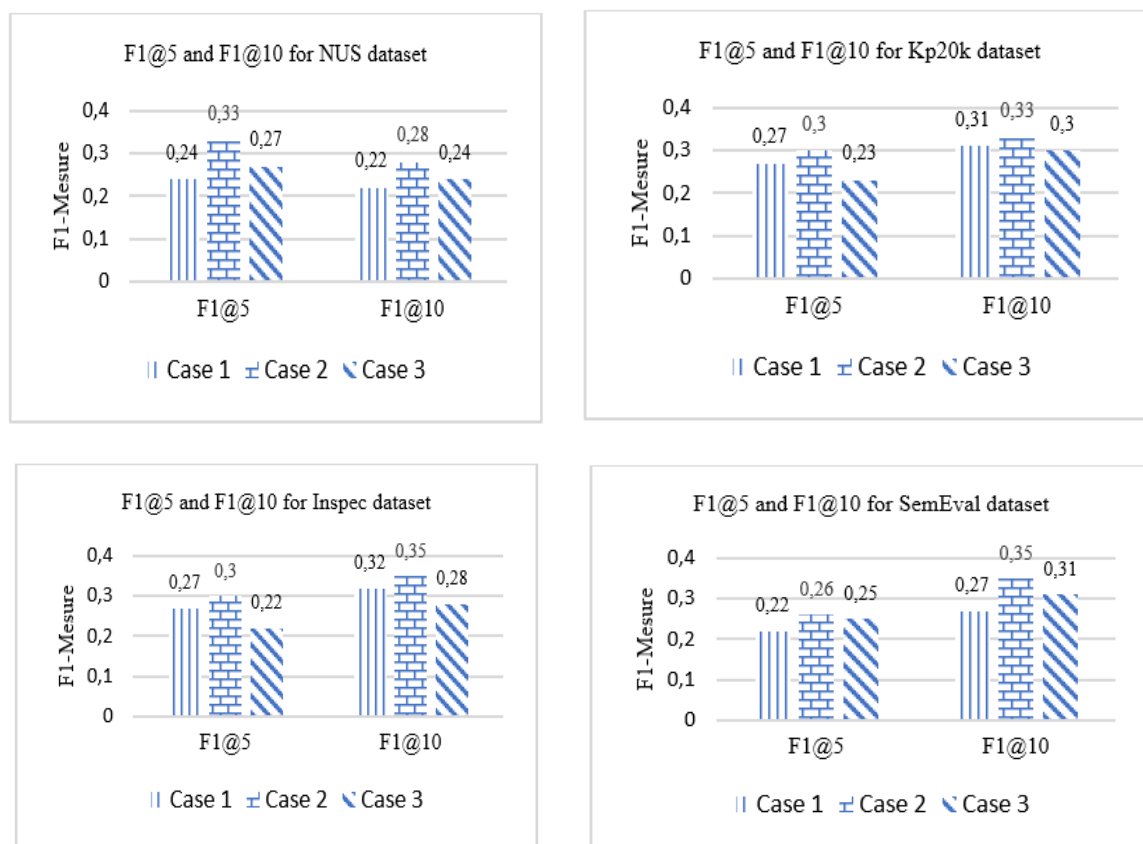


Figure 5. The result of F.Measure in different case of selection of paragraphs express the document

#### 4.4. Evaluation results

Our method underwent two stages of evaluation. The first was limited to evaluating the performance of present keyphrases extraction and comparing it with the performance of methods that use the embedding technique to extract keyphrases. The second stage involved evaluating the performance of absent keyphrases generation and comparing it with the methods for generating absent keyphrases.

##### 4.4.1. Present keyphrases extraction

To evaluate the performance of our method for present keyphrases extraction, we compared it with three methods that also adopt embedding techniques for keyphrase extraction. The first is EmbedRank [48] is an unsupervised method for extracting the present keyphrases. It embeds candidate keyphrases and the document using Sent2vec embedding technique [49]. The keyphrases are selected from among the candidate keyphrases that have the greatest cosine similarity to the document using the maximal margin relevance, to avoid repetition of extracting the same keyphrases. MDERank [28], an unsupervised method that uses BERT technique [50] to embed the document and its variants. The principle of MDERank is to create variants for the original document while masking some phrases in these variants. Semantic similarity is calculated between these variants and the original document. Masked phrases in the variant that achieve the least semantic similarity to the original document are of great importance to him. The third is KP-USE [51], which is an unsupervised method. It is based on dividing the document into five main parts. These parts and candidate phrases are embedded by the USE technique. The semantically similar phrases of these parts are keyphrases in the document.



Table 3 shows the results of the performance of our method compared to the performance of these methods in the dataset that we used. Each method extracted 5 keyphrases in the first phase and then 10 keyphrases in the second phase. Our method excels in datasets containing long texts. On the other hand, there is convergence in performance with other methods in datasets that contain short texts.

Table 3. KPEG performance for extracting present keyphrases compared with 3 methods

Method	NUS		KP20k		Inspec		SemEval	
	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10	F1@5	F1@10
EmbedRank	0.13	0.17	0.28	0.31	<b>0.31</b>	0.34	0.16	0.21
MDERank	0.15	0.18	<b>0.32</b>	<b>0.34</b>	0.28	0.32	0.17	0.20
KP-USE	0.07	0.08	0.25	0.28	0.22	0.26	0.15	0.20
KPEG	<b>0.33</b>	<b>0.28</b>	0.30	0.33	0.30	<b>0.35</b>	<b>0.26</b>	<b>0.35</b>

#### 4.4.2. Absent keyphrases generation

To measure how well our method generates absent keyphrases we select to compare its performance with that of two methods that generate absent keyphrases. The first is proposed in [23]. The method generates absent keyphrases using a model based on RNN encoder-decoder, enhanced by a copying mechanism [52] that enables the RNN to generate appropriate phrases from the source text. The second method is TG-Net [53], which considers that the title of the document has an essential role in the task of generating absent key phrases, since the title is used by TG-Net as an additional query in the input to the encoder-decoder, in addition to the document text, allows This form makes use of the information included in the title to create absent key phrases. Generating absent keyphrases is a very complex task. Where, we note that most methods are evaluated for their performance by generating ten or more keyphrases. So, we will evaluate our method based on its performance in generating 10 keyphrases in the first stage and then 50 keyphrases in the second stage.

Table 4 presents the results of the recall metric as shown in (5), for the generation of absent keyphrases for our method and comparing it with the CopyRNN and TG-Net methods, in which we note that our method is able to obtain a recall average of approximately 10 %, which is a fairly acceptable average, especially if we consider the percentage of the present keyphrases that over 60% of all keyphrases.

Table 4. KPEG performance for generating absent keyphrases compared with 2 methods

Method	NUS		KP20k		Inspec		SemEval	
	R@10	R@50	R@10	F1@50	R@10	F1@50	F1@10	F1@50
CopyRNN	0.06	0.12	0.13	0.21	0.05	0.10	0.04	0.07
TG-Net	0.08	0.12	<b>0.16</b>	<b>0.27</b>	0.06	<b>0.12</b>	0.05	0.08
KPEG	<b>0.07</b>	<b>0.13</b>	0.12	0.18	<b>0.06</b>	0.09	<b>0.07</b>	<b>0.10</b>

#### 4.5. Discussion

The unsupervised approach proposed in this paper combines the extraction of keyphrases present in the document, and the generation of absent keyphrases. To extract the present keyphrases, we exploited the USE embedding technique to select the paragraphs expressing the document, as we considered that the phrases similar to these paragraphs are present keyphrases. While we used the RNN encoder-decoder model to generate absent keyphrases. Instead of the source text, we used as an input in this model the paragraphs expressing the document in order to avoid generating phrases that are irrelevant from the document and reduce the complexity of the model. The list of absent keyphrases generated is filtered to avoid duplication of keyphrases. The keyphrases, whether present or absent, are ranked according to the score of their proximity to the paragraphs, to select the N first phrases as keyphrases of the document.

The results of our evaluation showed that our method performed well in extracting the present keyphrases compared to methods that used embedding techniques, especially in long documents. The evaluation in which we exploited four datasets also showed that our method has the ability to generate absent keyphrases, with a recall average over 10%. This result remains encouraging as we will improve it in the future by providing a larger dataset for training, especially for short texts. Most methods that generate absent keyphrases find it difficult to overcome the problem of phrase overlap and duplication. There are a number of solutions that have been proposed to overcome this problem, as [24] who proposed an automatic review to reduce duplicates. Zhao and Zhang [54] apply constraints to limit the generating of overlapped phrases. To overcome this problem, our method proposed Algorithm 1 that removes overlapped or duplicate phrases after extracting and generating keyphrases, whether they are present or absent.

## 5. CONCLUSION

This paper presents an unsupervised method that combines the extraction of the present keyphrases and the generation of the absent keyphrases. We exploited the USE embedding technique to extract the keyphrases from the expressive paragraphs of the document, while the absent keyphrases were generated by exploiting the RNN encoder-decoder where we used it as an input for the expressive paragraphs of the document. We evaluated our method on four datasets namely NUS and SemEval containing long documents and Inspec and KP20k containing short texts. The results showed the superiority of our method in extracting the present keyphrases compared to other methods, especially in long documents. Also, the results we obtained in generating the absent keyphrases remain encouraging. Our method proposed a new algorithm that reduces the problem of overlap and duplicate keyphrases. In the future we will improve the performance of our method of generating absent keyphrases by providing a larger training dataset, especially for short texts.

## REFERENCES




- [1] N. Nikzad-Khasmakhi *et al.*, “Phraseformer: multimodal key-phrase extraction using transformer and graph embedding,” *arxiv preprints*, Jun. 2021, doi: 10.48550/ARXIV.2106.04939.
- [2] E. Papagiannopoulou and G. Tsoumakas, “A review of keyphrase extraction,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 2, Mar. 2020, doi: 10.1002/widm.1339.
- [3] H. Tsukagoshi, R. Sasano, and K. Takeda, “Comparison and Combination of sentence embeddings derived from different supervision signals,” *arxiv preprints*, pp. 139–150, Feb. 2022, doi: 10.18653/v1/2022.starsem-1.12.
- [4] L. Ajalloua, K. Najmani, A. Zellou, and E. H. Benlahmar, “Doc2Vec, SBERT, InferSent, and USE Which embedding technique for noun phrases?,” in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2022*, Mar. 2022, pp. 1–5, doi: 10.1109/IRASET52964.2022.9738300.
- [5] D. Cer *et al.*, “Universal sentence encoder for English,” in *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, Mar. 2018, pp. 169–174, doi: 10.18653/v1/d18-2029.
- [6] S. Arora, Y. Liang, and T. Ma, “A simple but tough-to-beat baseline for sentence embeddings,” *International Conference on Learning Representations (2017)*, 2017, pp. 1-16.
- [7] Z. Yang, C. Zhu, and W. Chen, “Parameter-free sentence embedding via orthogonal basis,” in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 638–648, doi: 10.18653/v1/d19-1059.
- [8] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised learning of universal sentence representations from natural language inference data,” *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 670–680, May 2017, doi: 10.18653/v1/d17-1070.
- [9] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, Aug. 2019, pp. 3982–3992, doi: 10.18653/v1/d19-1410.
- [10] A. Vaswani *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems*, Jun. 2017, pp. 5999–6009.
- [11] O. Terrada, S. Hamida, B. Cherradi, A. Raihani, and O. Bouattane, “Supervised machine learning based medical diagnosis support system for prediction of patients with heart disease,” *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 5, pp. 269–277, 2020, doi: 10.25046/AJ050533.
- [12] O. Asmae, R. Abdelhadi, C. Bouchaib, S. Sara, and K. Tajeddine, “Parkinson’s disease identification using KNN and ANN algorithms based on voice disorder,” in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2020*, Apr. 2020, pp. 1–6, doi: 10.1109/IRASET48871.2020.9092228.
- [13] B. Cherradi, O. Terrada, A. Ouhmida, S. Hamida, A. Raihani, and O. Bouattane, “Computer-aided diagnosis system for early prediction of atherosclerosis using machine learning and K-fold cross-validation,” in *2021 International Congress of Advanced Technology and Engineering, ICOTEN 2021*, Jul. 2021, pp. 1–9, doi: 10.1109/ICOTEN52080.2021.9493524.
- [14] S. Siddiqi and A. Sharan, “Keyword and keyphrase extraction techniques: a literature review,” *International Journal of Computer Applications*, vol. 109, no. 2, pp. 18–23, Jan. 2015, doi: 10.5120/19161-0607.
- [15] C. Caragea, F. Bulgarov, A. Godea, and S. Das Gollapalli, “Citation-enhanced keyphrase extraction from research papers: a supervised approach,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 1435–1446, doi: 10.3115/v1/d14-1150.
- [16] C. Florescu and W. Jin, “Learning feature representations for keyphrase extraction,” in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, Jan. 2018, pp. 8077–8078, doi: 10.1609/aaai.v32i1.12144.
- [17] S. R. El-Beltagy and A. Rafea, “KP-Miner: A keyphrase extraction system for English and Arabic documents,” *Information Systems*, vol. 34, no. 1, pp. 132–144, Mar. 2009, doi: 10.1016/j.is.2008.05.002.
- [18] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, “YAKE! collection-independent automatic keyword extractor,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10772 LNCS, 2018, pp. 806–810.
- [19] M. Won, B. Martins, and F. Raimundo, “Automatic extraction of relevant keyphrases for the study of issue competition,” in *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing*, 2019.
- [20] R. Mihalcea and P. Tarau, “TextRank: Bringing order into texts,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2004*, 2004, pp. 404–411.
- [21] F. Boudin, “Unsupervised keyphrase extraction with multipartite graphs,” in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Mar. 2018, vol. 2, pp. 667–672, doi: 10.18653/v1/n18-2105.
- [22] Y. Yu and V. Ng, “WikiRank: improving keyphrase extraction based on background knowledge,” in *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, Mar. 2019, pp. 3723–3727, [Online]. Available: <http://arxiv.org/abs/1803.09000>.

- [23] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, Apr. 2017, vol. 1, pp. 582–592, doi: 10.18653/v1/P17-1054.
- [24] 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, EMNLP 2018*, 2018, pp. 4057–4066, doi: 10.18653/v1/d18-1439.
- [25] R. A. Al-Zaidy, C. Caragea, and C. Lee Giles, "Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents," in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, 2019, pp. 2551–2557, doi: 10.1145/3308558.3313642.
- [26] 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, doi: 10.1109/ACCESS.2020.2965087.
- [27] J. R. Asl and J. M. Banda, "GLEAKE: global and local embedding automatic keyphrase extraction," *arxiv preprints*, May 2020, [Online]. Available: <http://arxiv.org/abs/2005.09740>.
- [28] L. Zhang et al., "MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction," *Findings of the Association for Computational Linguistics: ACL 2022*, Oct. 2022, pp. 396–409, doi: 10.18653/v1/2022.findings-acl.34.
- [29] M. Crawshaw, "Multi-task learning with deep neural networks: a survey," *arxiv preprints*, Sep. 2020, [Online]. Available: <http://arxiv.org/abs/2009.09796>.
- [30] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 632–642, doi: 10.18653/v1/d15-1075.
- [31] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 168–177, doi: 10.1145/1014052.1014073.
- [32] B. Pang and L. Lee, "A sentimental analysis: sentiment analysis using subjectivity summarization based on minimum cuts," *ACL '04: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004, doi: 10.3115/1218955.1218990.
- [33] G. Rabby, S. Azad, M. Mahmud, K. Z. Zamli, and M. M. Rahman, "TeKET: a tree-based unsupervised keyphrase extraction technique," *Cognitive Computation*, vol. 12, no. 4, pp. 811–833, Jul. 2020, doi: 10.1007/s12559-019-09706-3.
- [34] T. D. Nguyen and M.-Y. Kan, "Keyphrase extraction in scientific publications," in *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, 2008, pp. 317–326, doi: 10.1007/978-3-540-77094-7\_41.
- [35] L. Ajallouda, O. Hourrane, A. Zellou, and E. H. Benlahmar, "Toward a new process for candidate key-phrases extraction," in *Lecture Notes in Networks and Systems*, vol. 455 LNNS, 2022, pp. 466–474, doi: 10.1007/978-3-031-02447-4\_48.
- [36] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, vol. 1, pp. 1681–1691, doi: 10.3115/v1/p15-1162.
- [37] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, 2005, pp. 115–124, doi: 10.3115/1219840.1219855.
- [38] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, no. 2–3, pp. 165–210, May 2005, doi: 10.1007/s10579-005-7880-9.
- [39] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2013, pp. 1631–1642.
- [40] Y. Yoo, T. S. Heo, Y. Park, and K. Kim, "A novel hybrid methodology of measuring sentence similarity," *Symmetry*, vol. 13, no. 8, p. 1442, Aug. 2021, doi: 10.3390/sym13081442.
- [41] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: encoder–decoder approaches," in *Proceedings of SSTS 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Sep. 2014, pp. 103–111, doi: 10.3115/v1/w14-4012.
- [42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, no. January, pp. 3104–3112, Sep. 2014, [Online]. Available: <http://arxiv.org/abs/1409.3215>.
- [43] W. Chen, H. P. Chan, P. Li, and I. King, "Exclusive hierarchical decoding for deep keyphrase generation," *arxiv preprints*, pp. 1095–1105, Apr. 2020, doi: 10.18653/v1/2020.acl-main.103.
- [44] W. U. Ahmad, X. Bai, S. Lee, and K. W. Chang, "Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 1389–1404, doi: 10.18653/v1/2021.acl-long.111.
- [45] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the 2003 conference on Empirical methods in natural language processing -*, 2003, vol. 10, pp. 216–223, doi: 10.3115/1119355.1119383.
- [46] S. N. Kim, O. Medelyan, M. Y. Kan, and T. Baldwin, "SemEval-2010 Task 5: Automatic keyphrase extraction from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010, pp. 21–26.
- [47] F. Liu, X. Huang, W. Huang, and S. X. Duan, "Performance evaluation of keyword extraction methods and visualization for student online comments," *Symmetry*, vol. 12, no. 11, pp. 1–20, Nov. 2020, doi: 10.3390/sym12111923.
- [48] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi, "Simple unsupervised keyphrase extraction using sentence embeddings," in *CoNLL 2018 - 22nd Conference on Computational Natural Language Learning, Proceedings*, 2018, pp. 221–229, doi: 10.18653/v1/k18-1022.
- [49] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, vol. 1, pp. 528–540, doi: 10.18653/v1/n18-1049.
- [50] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Oct. 2019, vol. 1, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
- [51] L. Ajallouda, F. Z. Fagroud, A. Zellou, and E. Ben Lahmar, "KP-USE: An unsupervised approach for key-phrases extraction from documents," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 283–289, 2022, doi: 10.14569/IJACSA.2022.0130433.
- [52] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, Mar. 2016, vol. 3, pp. 1631–1640, doi: 10.18653/v1/p16-1154.




- [53] W. Chen, Y. Gao, J. Zhang, I. King, and M. R. Lyu, "Title-guided encoding for keyphrase generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Aug. 2019, pp. 6268–6275, doi: 10.1609/aaai.v33i01.33016268.
- [54] J. Zhao and Y. Zhang, "Incorporating linguistic constraints into keyphrase generation," in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020, pp. 5224–5233, doi: 10.18653/v1/p19-1515.

## BIOGRAPHIES OF AUTHORS



**Lahbib Ajallouda**    PhD Student at Computer Science and Systems Analysis School (ENSIAS), Mohamed V University, Rabat, Morocco. His research interests are primarily in the area of internet of things, search engines, cloud computing, and machine learning, where he is the author/co-author of over 9 research publications. He can be contacted at email: lahbib\_ajallouda@um5.ac.ma.



**Ahmed Zellou**    Received his Ph.D. in Applied Sciences at 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 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 72 research publications. He can be contacted at email: ahmed.zellou@um5.ac.ma.