KP-USE: An Unsupervised Approach for Key-Phrases Extraction from Documents

Lahbib Ajallouda[1], Fatima Zahra Fagroud[2], Ahmed Zellou[3], Elhabib Ben Lahmar[4]
SPM Research Team, ENSIAS, ENSIAS, Mohammed V University, Rabat, Morocco[1, 3]
LTIM–FSBM, FSBM Hassan II University, Casablanca, Morocco[2, 4]

***Abstract*—Automatic key-phrase extraction (AKE) is one of the most popular research topics in the field of natural language processing (NLP). Several techniques were used to extract the key-phrases: statistical, graph-based, classification algorithms, deep learning, and embedding techniques. AKE approaches that use embedding techniques are based on calculating the semantic similarity between a vector representing the document and the vectors representing the candidate phrases. However, most of these methods only give acceptable results in short texts such as abstracts paper, but on the other hand, their performance remains weak in long documents because it is represented by a single vector. Generally, the key phrases of a document are often expressed in certain parts of the document as, the title, the summary, and to a lesser extent in the introduction and the conclusion, and not of the entire document. For this reason, we propose in this paper KP-USE. A method extracts key-phrases from long documents based on the semantic similarity of candidate phrases to parts of the document containing key-phrases. KP-USE makes use of the Universal Sentence Encoder (USE) as an embedding method for text representation. We evaluated the performance of the proposed method on three datasets containing long papers, namely, NUS, Krapivin2009, and SemEval2010, where the results showed its performance outperforms recent AKE methods which are based on embedding techniques.**

Keywords—Key-phrase extraction; natural language processing; semantic similarity; embedding technique; universal sentence encoder

I. Introduction

The explosive growth in the number of digital documents has prompted researchers to find effective ways to analyze and summarize all these documents [1]. AKE is one of the best document content analysis solutions. Researchers have used several techniques to improve the performance of key-phrase extraction [2]. Sentence embedding [3] is a recent technique that has been used to select key-phrases using a similarity measure [4]. Most of these methods perform less well, especially in long documents, because the document contains a large amount of information, which negatively affects its representation by a single vector. In general, most documents contain the title, abstract, introduction, and conclusion. These are the parts likely to contain key-phrases. The calculation of the similarity between the candidate phrases and the document must therefore take this factor into account.

The objective of this article is to propose KP-USE, a new, unsupervised method for key-phrases extraction from documents. KP-USE divides the document into five main parts: title, abstract, introduction, body, and conclusion. It represents each part by a USE technique [5] as a sentence embedding technique, in which the semantic similarity between candidate key-phrases and document is based on the phrases proximity to these parts, giving preference to parts that often contain key-phrases.

The rest of the article is organized as follows. In Section 2, we discuss related works. The USE sentence embedding technique is presented in Section 3, and then we present the proposed method for key-phrase extraction in Section 4. We empirically evaluate KP-USE in Section 5. Finally, we conclude the article in Section 6.

II. Related Work

In this section, we will talk about the most important related work to automatic key-phrases extraction, and in addition, we will introduce sentence embedding techniques.

A. Key-phrases Extraction Approaches

Over the past twenty years, many automatic methods of key-phrase extraction have been proposed. Siddiqi et al. in [6] classify these approaches into three sets, supervised, Semi-Supervised and unsupervised methods. Generally, the extraction by supervised approaches can be considered as a binary classification problem, where the candidate phrases are classified either as a key-phrase or a non-key phrase. While unsupervised methods rely on ordering candidate phrases based on calculating the score from one or more weights.

While Papagiannopoulou et al. in [2] suggested classifying these methods according to the approved techniques, methods based on statistics such as [7], [8], [9]. Graph-based methods such as [10], [11], [12]. Methods that use binary classification such as [13], [14]. Methods based on deep learning as [15],[16], [17] and methods based on embedding techniques such as [18], [19], [20].

B. Sentence Embedding Technics

Sentence embedding is an efficient way to convert textual data into fixed-length multidimensional vectors. Sentence embedding methods can be classified into two categories: (i) non-parameterized methods such as SIF [21], uSIF [22], and GEM [23] which rely on a simple technique, by encoding the words that make up the sentence and averaging the resulting vectors as a vector representing the sentence. However, this technique neglects information about word order and sentence semantics. (ii) parameterized more complex methods and generally perform better than unparameterized models [24].

The popularity of encoders has led to a great evolution of these methods. Conneau et al. propose in [25] InferSent is a technique based on a supervised RNN model predicting the semantic relations between pairs of sentences. Universal Sentence Encoder [5] is a trained and optimized technique for short sentences and paragraphs, it is based on two encoder models, Transformer and Deep Average Network (DAN). Subramanian et al. propose in [26] a more complex model for learning sentence embeddings in a multitasking configuration. Reimers and Gurevych propose in [27] Sentence-BERT, which also uses a Siamese network to create BERT-based sentence embeds [28]. Generally, Parameterized methods outperform non-parameterized ones on many tasks, but they are computationally expensive and require more training time.

III. Universal Sentence Encoder

Universal Sentence Encoder (USE) is a sentence embedding model that encodes a sentence or paragraph into a 512-dimensional vector. This vector encodes the meaning of the sentence and can therefore be used as input for NLP tasks such as document classification, key-phrase extraction, and textual similarity analysis.

A. *USE Process*

The idea of USE is to encode sentences into 512-dimensional vectors, via an encoder. These vectors are used in many NLP tasks. Depending on the errors made in these tasks, USE again reproduces vectors for these sentences. Fig. 1 shows the USE process.

The USE process begins with a tokenization operation, where sentences are converted to lowercase and tokenized into tokens. In the second step, USE encodes the sentence.

B. *Encoder*

USE offers two architectures for encoding the sentence. The first is based on a transformer consisting of six layers. Each layer has a feedback network preceded by a self-attention module that takes into account word order and context when generating each word representation. Fig. 2 shows USE architecture with transformer encoder.

The second is based on Deep Average Network (DAN) proposed by Iyyer et al in [29], where the vectors of the words in the sentence are averaged. The resulting vector is then passed through a 4-layer deep neural network (DNN) to obtain a vector of 512 dimensions. Fig. 3 shows USE architecture with DAN.

Generally, the result obtained through the transformer is very precise but requires more calculation time. It is therefore difficult to use for long texts, whereas DAN generates text encoding in less time but with less accuracy than a transformer.

C. *Multi-task Learning*

After the encoding of the sentences, either by transformers or by DAN. USE relies on multi-task learning (MTL) [30] to exploit commonalities and differences across tasks. This improves its learning efficiency and the accuracy of its predictions in the vector representation of sentences. Also, the authors of USE exploited a set of sources as training data. In addition to Stanford Natural Language Inference (SNLI) corpus [31], some web resources such as Wikipedia, web news, question and answer web pages, and discussion forums were also used.

Once USE has been trained, it can be used to represent any text, whether it is a phrase, a sentence, or a paragraph, with a 512-dimensional vector.

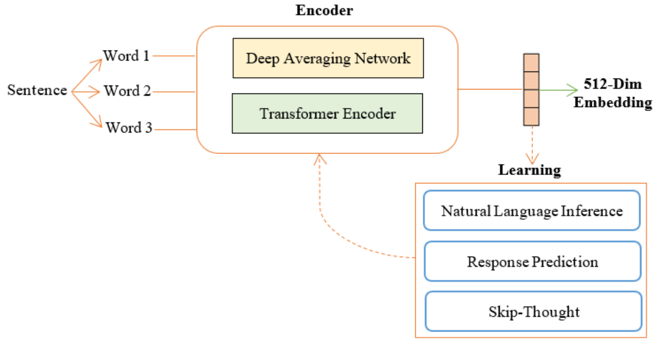

Fig. 1. Universal Sentence Encoder Process.

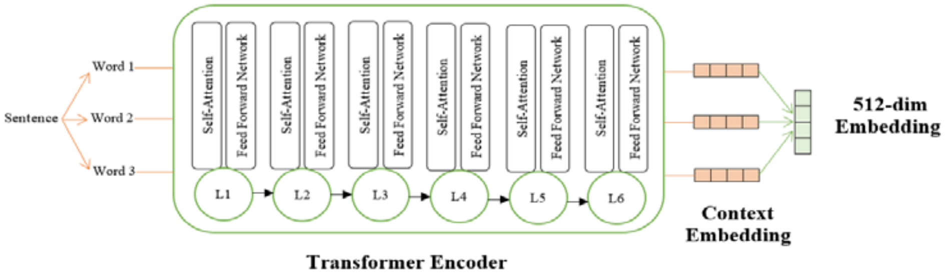

Fig. 2. USE Architecture with Transformer Encoder.

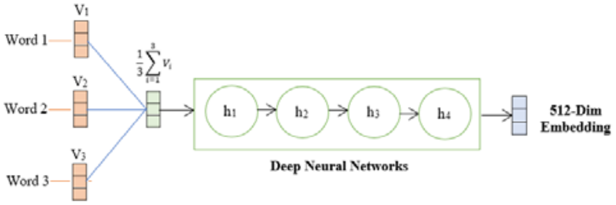

Fig. 3. USE Architecture with Deep Average Network.

D. Evaluation of USE

To show that embedding USE provides a better text representation. The authors used semantic similarity in the different tasks of SentEval [32]. They also preferred using angular distance (Formula 2) as a measure of similarity rather than cosine (Formula 1) because, in their opinion, it gave better results.

$$sim(p,q) = \frac{\sum p_i q_i}{\sqrt{\sum p_i^2}\sqrt{\sum q_i^2}} \qquad (1)$$

$$ang\_sim(p,q) = 1 - \frac{1}{\pi} arccos\big(sim(p,q)\big) \qquad (2)$$

- DataSet: Binary classification and sentence similarity tasks in SentEval are used to evaluate the quality of sentence representations. Table I shows the datasets used to evaluate the performance of USE. These datasets are used by USE authors to evaluate the two USE models, Transformer and DAN.
- Results: The experimental results showed the performance of USE in many tasks, whether in the transformer model or the DAN model. In Table II, we show the results of the two models for different datasets.

We also observe that encoding by transformer generally works better than encoding by DAN. On the other hand, the USE authors point out that the complexity of the transformer-based model is O(n^2), while the DAN model is O(n). Therefore, the transformer model is slower than the DAN model; especially with increasing sentence length.

TABLE I. DATASETS USED TO EVALUATE THE PERFORMANCE OF USE

Dataset	Number of samples	Task
Text REtrieval Conference (TREC) [33]	6 000	Question and answering
Customer Reviews (CR) [34]	4 000	Product reviews
Subjectivity Summarization (SUBJ) [35]	10 000	Subjectivity/Objectivity
Multi-Perspective Question and Answering (MPQA) [36]	11 000	Opinion polarity
Movie Reviews (MR) [37]	11 000	Sentiment
Stanford Sentiment Analysis (SST) [38]	67 300	Sentiment

TABLE II. USE PERFORMANCE ACCORDING TO EACH ENCODING MODEL

Dataset	USE DAN	USE Transformer
TREC	91.19	92.51
CR	80.97	87.43
SUBJ	92.65	93.87
MPQA	85.38	86.98
MR	74.45	81.44
SST	77.62	85.38

IV. KP-USE APPROACH

In this section, we propose KP-USE, which is a new method for key-phrases extraction from documents.

A. KP-USE Process

The KP-USE process presented in Fig. 4 shows that our method consists of five main steps:

Step 1: Extraction of candidate key-phrases;

Step 2: Split the document into five main sections;

Step 3: Vector representation of main sections and candidate key-phrases by USE;

Step 4: Calculate the score of each candidate key-phrase;

Step 5: Sorting the candidate key-phrases according to their score and extracting the phrases with the best score as key-phrases.

B. Candidate Key-phrases

The process of candidate key-phrases extraction (Fig. 5) begins with removing non-text data from the document, as well as converting all text to lowercase, and translating foreign words from the text into the language being studied. Then, the tokenization technique is used to convert the cleaned text into an array of tokens, to determine the grammatical category of each word using the Part-Of-Speech Tagging (POST) technique.

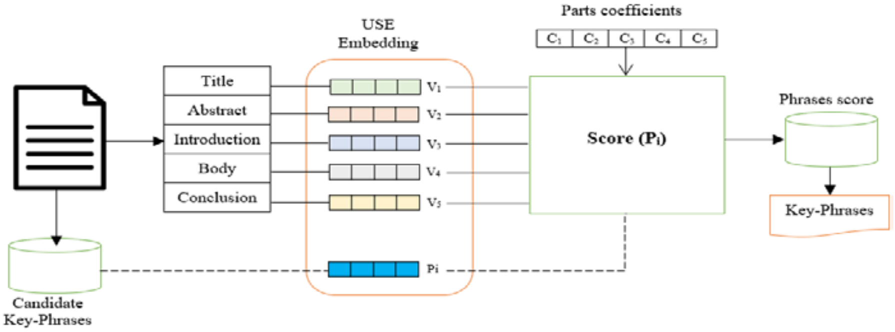

Fig. 4. The KP-USE Process.

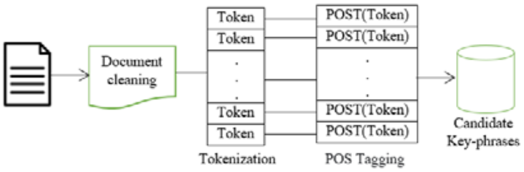

Fig. 5. Candidate Key-phrases Process.

Several AKE approaches have noted that key-phrases are noun phrases composed of one or more words such as [1], [39], and [40]. According to [19], the gerund form of a verb (VBG) can be used as a noun and the past participle form of a verb (VBN) can be considered as an adjective in the composition of key sentences. Therefore, (NN.*|JJ.*|VBN|VBG)*(NN.*|VBG) is the proposed pattern for candidate key-phrases extraction.

C. Document Fraction

Through our analysis of more than 50 scientific papers, we noticed that most of the key-phrases are semantically similar to the title of the article and are often mentioned in the abstract and to a lesser extent in the introduction and the conclusion. Based on this result (Table III), we propose to split the document into five parts, namely title, abstract, introduction, and conclusion, while the fifth part comprises the rest of the document. To favor the parts similar to the key phrases. In Table III, we propose for each part a proximity coefficient which will be used when calculating the similarity between the candidate phrases and the document.

TABLE III. THE PROXIMITY COEFFICIENTS FOR EACH PART

Section	Average Similarity	Coefficient
Title	0.61	3
Abstract	0.68	3
Introduction	0.42	2
Body	0.14	1
Conclusion	0.35	2

Some documents may not have one of the five parts. In this case, KP-USE considers the proximity coefficient of the non-existent part to be zero. Therefore, there will be no effect on the calculation of the similarity between the candidate phrases and the document. Thus, we will be able to apply KP-USE to any document.

D. Sentence Embedding

Our method proposes to extract key-phrases from the document based on the calculation of semantic similarity between the candidate key-phrases and the five parts of the document that have been identified in Table III. Before calculating the similarity, we must first represent the phrases candidate and the five parts by vectors. For this, we exploit USE as an embedding technique to obtain vectors of 512 dimensions. We test USE for encoding models, Transformer and DAN, to see which works best.

E. Candidate Key-phrase Score

The score of any candidate key-phrase is expressed by its semantic similarity with the document. This score is calculated based on the similarity of the candidate phrase to each of the five parts of the document, which is calculated by Formula 1. To exploit the proximity coefficient of each part of the candidate phrase, KP-USE uses Formula 3 to calculate the score for each candidate key-phrase.

$$Score(P_i) = \frac{1}{\sum_{j=1}^{5} C_j} \sum_{k=1}^{5} sim(V_k, U_i) \times C_k \quad (3)$$

P_i: The candidate key-phrase i.

U_i: the vector that represents phrase i.

V_k: The vector that represents part k.

C_k: Coefficient of part k.

C_j: Coefficient of part j.

F. Key-phrases Extraction

After calculating the score for each candidate key-phrase, KP-USE ranks these statements in descending order based on the score. KP-USE allows the user to choose the number of key-phrases to extract. The highest-rated phrases as key-phrases in the document.

V. Experimental Evaluation

In this section, we will present the datasets that were used to evaluate KP-USE. Additionally, we will verify which of the two models exploited in USE performs better in AKE by evaluating their performance using the precision, recall, and F1-score metrics. Then, we compare the performance of our method with other AKE methods based on embedding techniques.

A. Datasets

To evaluate the performance of our method, we used three datasets that are considered the most widely used in evaluating key-phrase extraction approaches, which are.

- NUS [41], is a set of scientific data, consisting of 211 conference papers where each paper contains between 4 and 12 pages. Keywords were identified by the volunteer students where each student was asked to read three articles and extract the keywords.
- Krapivin2009 [42]: It is considered one of the largest datasets in terms of number documents, containing 2,304 research articles in the field of computer science. The keywords of the articles were identified by the authors and checked by the reviewers.
- Semeval-2010 [43]: This is one of the most widely used data sets for evaluating keyword extraction approaches. It consists of 244 scientific articles belonging to the field of computer science, where the number of article pages varies from 6 to 8 pages. The keywords for each article are defined by the authors and professional editors. Table IV shows the statistics for the three datasets.

The most important feature of the datasets used is that they contain scientific papers consisting of 4 and 15 pages, i.e. each paper contains the five parts we have discussed in Table III will confirm the credibility of the results we will obtain.

TABLE IV. DATASETS USED TO EVALUATE KP-USE

Dataset	**Type of doc**	**Docs**	**Language**	**Tokens per Doc**	**KP per Doc**	**Annotation**
NUS	Full paper	211	EN	8398.30	11.00	Reader
Krapivin2009	Full paper	2304	EN	8040.74	6.34	Author
SemEval2010	Full paper	244	EN	7961.20	14.70	Author/Reader

B. Evaluation Metrics

Several evaluation metrics are used in key-phrase extraction [44]. Researchers generally prefer three metrics, namely precision, recall, and F1.score because of their validity and ease of use. These are the same metrics we will use to evaluate KP-USE.

- Precision: To evaluate the precision of the approach, we use this metric. Its value is expressed as the proportion of correctly extracted key-phrases compared to the total number of extracted key-phrases. To calculate the precision, we use formula 4.

$$Precision = \frac{Correct\ KeyPhrases}{Extracted\ KeyPhrases} \tag{4}$$

- Recall: To evaluate the completeness of key-phrase extraction, we use the recall metric, which expresses the proportion of correctly extracted key-phrases among the author's or reader's selected key-phrases. To calculate the yield, we use formula 5.

$$Recall = \frac{Correctly\ Keyphrases}{Assigned\ Keyphrases} \tag{5}$$

- F1.Score: There is often an inverse interaction between precision and recall. When precision is high, recall is low. For this, the F1-score is used, to combine precision and recall. To calculate F1-score, we use formula 6.

$$F1.Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

C. Comparison of USE Models

To choose the appropriate encoding model for KP-USE, we experimented with extracting key phrases using both encoding models, Transformer and DAN. Table V shows the performance of KP-USE according to each model according to the F1-Score metric on the three datasets NUS, Krapivin, and Semeval2010 for top 5 (F1@5), top 10 (F1@10), and top 15 (F1@15).

From the results obtained, we find that KP-USE based on the Transformer model performs better than KP-USE based on the DAN model in all the datasets used. Although the complexity of the Transform model is higher than that of the DAN model, we prefer to use the Transformer-based model because it achieves promising results.

D. Performance Comparison

We compare KP-USE with three methods which are also based on embedding techniques namely EmbedRank [45] is an unsupervised method that uses the Sent2vec [46] embedding technique for the representation of phrases and documents. SIFRank [18] is an unsupervised key-phrase extraction method based on the SIF [21] and ELMo [47] embedding model. The third approach is MDERank [20], this is an unsupported approach. It implicitly embeds position and frequency offset information by encoding the document using BERT embedding technique [48].

In general, the results obtained by KP-USE remain acceptable compared to the performance of the other methods at the level of long documents. Table 6 present these results.

TABLE V. KP-USE PERFORMANCE ACCORDING TO TRANSFORMER AND DAN MODEL

Model	**NUS**			**Krapivin**			**Semeval**		
	F1@5	***F1@10***	***F1@15***	***F1@5***	***F1@10***	***F1@15***	***F1@5***	***F1@10***	***F1@15***
KP-USE Transformer	0.07	0.08	0.10	0.10	0.14	0.14	0.15	0.20	0.19
KP-USE DAN	0.03	0.06	0.10	0.08	0.11	0.11	0.12	0.16	0.17

TABLE VI. KP-USE PERFORMANCE COMPARED TO EMBEDRANK, SIFRANK, AND MDERANK

Approach	**NUS**			**Krapivin**			**Semeval**		
	F1@5	**F1@10**	**F1@15**	**F1@5**	**F1@10**	**F1@15**	**F1@5**	**F1@10**	**F1@15**
EmedRank	0.02	0.03	0.04	0.09	0.11	0.10	0.10	0.14	0.15
SIFRank	0.04	0.06	0.06	0.02	0.03	0.03	0.11	0.17	0.19
MDERank	**0.15**	**0.18**	**0.18**	**0.12**	0.13	0.13	0.13	0.17	**0.20**
KP-USE	0.07	0.08	0.10	0.10	**0.14**	**0.14**	**0.15**	**0.20**	0.19

E. Discussion

KP-USE is an unsupervised AKE method that takes advantage of the USE embedding technique to represent text vectorially. KP-USE relied on splitting the text to focus the search for key-phrases in the parts that might contain them, and this split also helps to improve the embedding of the document because instead of just being represented by a single vector, each part is represented by a vector. What also characterizes KP-USE is that the similarity calculation takes into account the parts of the document which often contain key-phrases, unlike other methods which consider all parts of the document to be of the same degree of importance, which makes KP-USE more efficient than these methods, especially in long documents. KP-USE can also be applied to short documents, where the importance coefficient for any part that does not exist in the document is zero, so the similarity calculation between candidate phrases and the document is not affected. KP-USE performance could be improved even more if we could develop it to be able to predict key-phrases that are not mentioned in the document.

VI. Conclusion

In this paper, we proposed KP-USE, an unsupervised method of extracting key-phrases from the document based on the USE embedding technique which uses the Transformer network model. Our method splits the document into five parts, namely title, abstract, introduction, body, and conclusion, to favor the parts likely to contain key-phrases while calculating the semantic similarity between the candidate key-phrases and the document. KP-USE was evaluated on three datasets continent of long documents, namely NUS, Krapivin, and Semeval 2010. F1-Score results showed that the performance of KP-USE was superior to the performance of unsupervised methods based on embedding techniques and on the calculation of semantic similarity to extract key-phrases. In the future, we will develop KP-USE to predict key-phrases not mentioned in the document.

Reference

[1] N. Nikzad-Khasmakhi, et al. "Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding." arXiv preprint arXiv:2106.04939 (2021).

[2] E. Papagiannopoulou, and G. Tsoumakas. "A review of keyphrase extraction." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10.2 (2020): e1339.

[3] H. Tsukagoshi, R. Sasano, and K. Takeda. "Comparison and Combination of Sentence Embeddings Derived from Different Supervision Signals." arXiv preprint arXiv:2202.02990 (2022).

[4] D. Buscaldi, et al. "Calcul de similarité entre phrases: quelles mesures et quels descripteurs?." 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes. ATALA; AFCP, 2020.

[5] D. Cer, et al. "Universal sentence encoder." arXiv preprint arXiv:1803.11175 (2018).

[6] S. Siddiqi, and A. Sharan. "Keyword and keyphrase extraction techniques: a literature review." International Journal of Computer Applications 109.2 (2015).

[7] S. R. El-Beltagy, and A. Rafea. "KP-Miner: A keyphrase extraction system for English and Arabic documents." Information systems 34.1 (2009): 132-144.

[8] R. Campos, et al. "Yake! collection-independent automatic keyword extractor." European Conference on Information Retrieval. Springer, Cham, 2018.

[9] M. Won, B. Martins, and F. Raimundo. "Automatic extraction of relevant keyphrases for the study of issue competition." Proceedings of the 20th international conference on computational linguistics and intelligent text processing. 2019.

[10] R. Mihalcea and P. Tarau. "Textrank: Bringing order into text." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

[11] F.Boudin. "Unsupervised keyphrase extraction with multipartite graphs." arXiv preprint arXiv:1803.08721 (2018).

[12] Y. Yu, and V. Ng. "Wikirank: Improving keyphrase extraction based on background knowledge." arXiv preprint arXiv:1803.09000 (2018).

[13] C. Caragea, F. Bulgarov, A. Godea and S. D. Gollapalli. "Citation-enhanced keyphrase extraction from research papers: A supervised approach." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[14] S. McIlraith, and K. Q. Weinberger. "Learning feature representations for keyphrase extraction." Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA. 2018.

[15] R. Meng et al. "Deep keyphrase generation." arXiv preprint arXiv:1704.06879 (2017).

[16] J. Chen, X. Zhang, Y. Wu, Z. Yan, and Z. Li. "Keyphrase generation with correlation constraints." arXiv preprint arXiv:1808.07185 (2018).

[17] R. Alzaidy, C. Caragea, and C. L. Giles. "Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents." The world wide web conference. 2019.

[18] 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 8 (2020): 10896-10906.

[19] J. R. Asl, and J. M. Banda. "GLEAKE: Global and Local Embedding Automatic Keyphrase Extraction." arXiv preprint arXiv:2005.09740 (2020).

[20] L. Zhang, et al. "MDERank: A Masked Document Embedding Rank Approach for Unsupervised Keyphrase Extraction." arXiv preprint arXiv:2110.06651 (2021).

[21] S. Arora, Y. Liang, and T. Ma. "A simple but tough-to-beat baseline for sentence embeddings." International conference on learning representations. 2017.

[22] K. Ethayarajh. "Unsupervised random walk sentence embeddings: A strong but simple baseline." Proceedings of The Third Workshop on Representation Learning for NLP. 2018.

[23] Z. Yang, C. Zhu, and W. Chen. "Parameter-free sentence embedding via orthogonal basis." arXiv preprint arXiv:1810.00438 (2018).

[24] B. Wang, and C. C. J. Kuo. "Sbert-wk: A sentence embedding method by dissecting bert-based word models." IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020): 2146-2157.

[25] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. "Supervised learning of universal sentence representations from natural language inference data." arXiv preprint arXiv:1705.02364 (2017).

[26] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal. "Learning general purpose distributed sentence representations via large scale multi-task learning." arXiv preprint arXiv:1804.00079 (2018).

[27] N. Reimers, and I. Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

[28] A. Vaswani, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[29] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. "Deep unordered composition rivals syntactic methods for text classification." Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). 2015.

[30] M. Crawshaw. "Multi-task learning with deep neural networks: A survey." arXiv preprint arXiv:2009.09796 (2020).

[31] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv:1508.05326 (2015).

[32] A. Conneau, and D. Kiela. "Senteval: An evaluation toolkit for universal sentence representations." arXiv preprint arXiv:1803.05449 (2018).

[33] X. Li, and D. Roth. "Learning question classifiers." COLING 2002: The 19th International Conference on Computational Linguistics. 2002.

[34] M. Hu, and B. Liu. "Mining and summarizing customer reviews." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. 2004.

[35] B. Pang, and L. Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." arXiv preprint cs/0409058 (2004).

[36] J. Wiebe, T. Wilson, and C. Cardie. "Annotating expressions of opinions and emotions in language." Language resources and evaluation 39.2 (2005): 165-210.

[37] B. Pang, and L. Lee. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales." arXiv preprint cs/0506075 (2005).

[38] R. Socher, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." Proceedings of the 2013 conference on empirical methods in natural language processing. 2013.

[39] G. Rabby, S. Azad, M. Mahmud, K. Z. Zamli, and M. M. Rahman. "Teket: a tree-based unsupervised keyphrase extraction technique." Cognitive Computation 12.4 (2020): 811-833.

[40] X. Shen, Y. Wang, R. Meng, and J. Shang. "Unsupervised Deep Keyphrase Generation." arXiv preprint arXiv:2104.08729 (2021).

[41] T. D. Nguyen, and M. Y. Kan. "Keyphrase extraction in scientific publications." International conference on Asian digital libraries. Springer, Berlin, Heidelberg, 2007.

[42] M. Krapivin, A. Autaeu, and M. Marchese. "Large dataset for keyphrases extraction." (2009).

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

[44] F. Liu, X. Huang, W. Huang, and S. X. Duan. "Performance evaluation of keyword extraction methods and visualization for student online comments." Symmetry 12.11 (2020).

[45] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. "Simple unsupervised keyphrase extraction using sentence embeddings." arXiv preprint arXiv:1801.04470 (2018).

[46] M. Pagliardini, P. Gupta, and M. Jaggi. "Unsupervised learning of sentence embeddings using compositional n-gram features." arXiv preprint arXiv:1703.02507 (2017).

[47] M. E. Peters, et al. "Deep contextualized word representations. arXiv 2018." arXiv preprint arXiv:1802.05365 12 (2018).

[48] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).