

Word centrality constrained representation for keyphrase extraction

Zelalem Gero and Joyce C Ho

Emory University

{zgero, joyce.c.ho}@emory.edu

Abstract

To keep pace with the increased generation and digitization of documents, automated methods that can improve search, discovery and mining of the vast body of literature are essential. Keyphrases provide a concise representation by identifying salient concepts in a document. Various supervised approaches model keyphrase extraction using local context to predict the label for each token and perform much better than the unsupervised counterparts. Unfortunately, this method fails for short documents where the context is unclear. Moreover, keyphrases, which are usually the gist of a document, need to be the central theme. We propose a new extraction model that introduces a centrality constraint to enrich the word representation of a Bidirectional long short-term memory. Performance evaluation on two publicly available datasets demonstrate our model outperforms existing state-of-the-art approaches. Our model is publicly available at https://github.com/ZHgero/keyphrases_centrality.git

1 Introduction

Keyphrase extraction is an important information extraction task that identifies single or multi-word linguistic units that concisely represent a document. They can also serve to provide a brief summary of the document content. Keyphrases are widely used in variety of natural language processing tasks such as document summarization (Bharti and Babu, 2017; Sarkar, 2014), query formulation (Jones and Staveley, 1999), text classification (Coenen et al., 2007), clustering (Hammouda et al., 2005), and recommendation systems (Naw and Hlaing, 2013). Keyphrases have become increasingly important for biomedical documents as there has been an exponential growth with over 32 million articles indexed by PubMed (NLM). Figure 1 shows a PubMed document with the author-specified keyphrases highlighted in blue.

Existing keyphrase extraction methods mainly fall either under a supervised or unsupervised approach. Common unsupervised approaches use word co-occurrence statistics to build graph-based ranking algorithms. Each word is mapped to a node and edges connect words that co-occur within a specified window size. Even though unsupervised approaches are desirable for datasets which do not have manually-labeled ground truth values, most such methods perform worse compared to the supervised counterparts.

The supervised approaches use classification to label every token as being part of a keyphrase or not by using features such as part-of speech tags, term-frequency inverse document frequency (tf-idf), and the position of the token in the document. Recently, supervised methods based on deep learning have been employed for keyphrase extraction. In Thomaidou and Vazirgiannis (2011) and Gollapalli et al. (2017), the authors posed the problem as a sequence labeling task and applied a Long Short-Term Memory network (LSTM) and conditional random fields (CRF) to tag each token in document as positive (i.e., part of a keyphrase) or negative. While these approaches achieve much better performance, they still suffer from a major limitation when applied on biomedical literature. The task of labelling each token does not consider how central the token is to the document contents. For Figure 1, the main theme of the keyphrases are genes associated with breast cancer. Thus, the document theme can be used as additional information to improve the keyphrase extraction performance.

To this end, we propose to address the problem of keyphrase extraction as a sequence labelling task with *an additional component to capture the centrality of each token*. We design a centrality layer built on top of a bidirectional LSTM (BiLSTM) layer to constrain each token with regards to the central theme of the document. The output dependencies are then modeled using a CRF layer. The

Protein tyrosine kinase (PTK) is one of the major signaling enzymes in the process of cell signal transduction, which catalyzes the transfer of ATP- γ -phosphate to the tyrosine residues of the substrate protein, making it phosphorylation, regulating cell growth, differentiation, death and a series of physiological and biochemical processes. Abnormal expression of PTK usually leads to cell proliferation disorders, and is closely related to tumor invasion, metastasis and tumor angiogenesis. At present, a variety of PTKs have been used as targets in the screening of anti-tumor drugs. Tyrosine kinase inhibitors (TKIs) compete with ATP for the ATP binding site of PTK and reduce tyrosine kinase phosphorylation, thereby inhibiting cancer cell proliferation. TKI has made great progress in the treatment of cancer, but the attendant acquired resistance is still inevitable, restricting the treatment of cancer. In this paper, we summarize the role of PTK in cancer, TKI treatment of tumor pathways and TKI acquired resistance mechanisms, which provide some reference for further research on TKI treatment of tumors.

Figure 1: An example document from PubMed with author-provided keyphrases in blue.

contributions of our work are:

- Introducing a centrality constraint layer to better capture the main theme of the document and how strongly each token is related to the main theme.
- Thorough evaluation of the centrality layer using an ablation study on biomedical and general domain abstracts.

The next section presents a brief description of the related work. The proposed keyphrase extraction method is introduced in Section 3. Sections 4, and 5 present experimental results and conclusion respectively.

2 Related Work

Keyphrase extraction methods mainly take either supervised or unsupervised approach. Unsupervised approaches generate candidates and rank using features such as tf-idf and topic proportions (Barker and Cornacchia, 2000; Liu et al., 2009b), graph-based centrality measures (Grineva et al., 2009; Wan and Xiao, 2008), topic modeling (Liu et al., 2009a; Teneva and Cheng, 2017), and document’s citation network (Gollapalli and Caragea, 2014). Unsupervised, graph-based methods build a graph from the input document where all the candidate keyphrases are nodes and the connection

between each candidate is represented by edges. A graph-based ranking method then determines the weights for each node based on the relatedness between the candidates. Alternatively, topic-based approaches cluster candidate keyphrases into topics in the document so that all the topics in the input document are represented by the selected keyphrases. Recently (Sun et al., 2020) proposed a sentence embedding model named SIFRank that uses autoregressive pre-trained language model to extract keyphrases from short documents. Yet unsupervised methods often fail to achieve state-of-the-art performance.

Under the supervised approach, the keyphrase extraction problem is treated as a binary classification task (Alzaidy et al., 2019; Turney, 2000, 2002), where learning algorithms such as support vector machines (Witten et al., 2005; Jiang et al., 2009) and maximum entropy (Kim and Kan, 2009; Yih et al., 2006) are used. Supervised keyphrase extraction can also be posed as a ranking problem between candidates (Witten et al., 2005). The candidates keys are extracted using statistical features (tf-idf, number of occurrences, first occurrence of the key) and structural features (part of speech tags).

Deep learning based models have also been used for keyphrase extraction. Word embeddings are used to measure the relatedness between words in graph-based models (Wang et al., 2014). Zhang et al. (2016) used a Recurrent Neural Network (RNN) based approach to identify keyphrases in Twitter data. The model addresses the problem as sequence labeling for very short text, where a joint-layer RNN is used to capture the semantic dependencies in the input sequence. Alzaidy et al. (2019) employed a LSTM-CRF architecture to model keyphrase extraction as a sequence labelling task to learn the labels of the entire input sequence. Santosh et al. (2020) extended the LSTM-CRF to utilize BiLSTM and incorporated an attention mechanism to retrieve additional information from other sentences within the same document. Sahrawat et al. (2020) evaluated the effect of various pre-trained word embeddings for the BiLSTM-CRF architecture in extracting keyphrases from benchmark datasets and found contextual embeddings offered better performance. While these models offer better performance, they fail to capture the centrality of the keyphrases which represent a salient feature of the document.

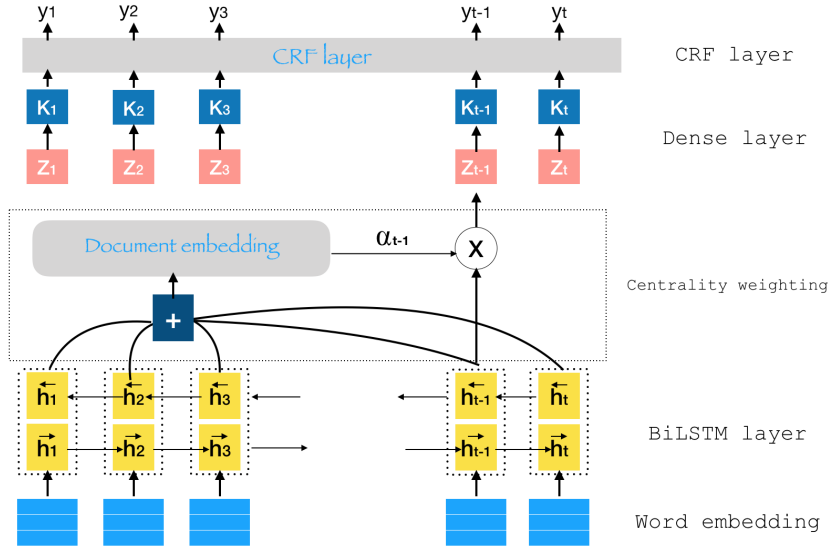


Figure 2: Our model architecture with the BiLSTM, centrality weighting, and CRF layer.

3 Methodology

The keyphrase extraction task is formulated as a sequence labelling task. Given a document $X = w_1, w_2, \dots, w_t$ where w_i is the i^{th} word and t is the number of words in the document, we predict the labels $y = y_1, y_2, \dots, y_t$ where each label y_i is whether word w_i is a keyphrase or not.

3.1 Word Embedding Layer

Each word in the document is represented by pre-trained low-dimensional vector representations. Any pre-trained vector representation can be used, and we experiment with various pre-trained embeddings such as GloVe (Pennington et al., 2014), BERT (Devlin et al., 2019) and BioBERT (Lee et al., 2020). The impact of each embedding type is discussed in the experiments section.

3.2 BiLSTM Layer

This layer is used to encode each document to obtain the local contextual representation. A forward and backward LSTMs are used to read the input sequence from left to right, $\vec{h}_1, \vec{h}_2, \dots, \vec{h}_t$, and right to left, $\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_t$, respectively. The outputs from the two directions are concatenated and summed for the final hidden state representation of the document, $H = [\sum_{i=1}^t \vec{h}_i, \sum_{i=1}^t \overleftarrow{h}_i]$.

3.3 Centrality Weighting Layer

Sequence labelling is commonly used for other token encoding tasks such as Named Entity Recognition (NER) where the task is to determine whether a token is a named entity or not. However, keyphrase

extraction is different from other sequence labelling tasks (for example NER) in that the tokens should capture the main gist of the document. This is in contrast to NER where the importance of the token is irrelevant as long as it is a named entity. To incorporate the idea of centrality, we use the similarity between each token and the document embedding, H , to bias the model towards tokens which are central (i.e., similar) to the document.

For words $\{w_1, w_2, \dots, w_t\}$ in a document D , we compute the centrality weight for each word $\alpha_1, \alpha_2, \dots, \alpha_t$. Each α_i is calculated as the cosine similarity between the document vector (H) and each word (w_i). This is then used to weight the document vector when concatenating with each word’s representation from the BiLSTM.

The output representation, z_i for each word is then the centrality weight, α_i multiplied by the output of the biLSTM, $z_i = [\alpha_i \vec{h}_i, \alpha_i \overleftarrow{h}_i]$. A dense layer is then used to transform the output representation, $k_i = f(z_i)$.

3.4 Conditional Random Fields (CRF)

The obtained contextual representations of each word, k_i are given as input sequence to a CRF layer. CRFs are widely used to model sequence labelling tasks (Lafferty et al., 2001). Given the input document as sequence of tokens, CRF produces a probability distribution over the output label sequence using the dependencies among the labels of the entire input sequence. This formulation considers the correlations between neighboring labels and allows joint decoding for the best sequence of

Table 1: Datasets used for experiments

Dataset	PubMed	INSPEC
Tot. documents	2532	500
Tot. # of tokens	654389	67200
Tot. # of keyphrases	31871	4912
Avg. # of keyphrases	12.5	9.8

labels for the input sequence, rather than decoding each label independently. Moreover, by utilizing two different labels for the keyphrase to denote the beginning (t_B) and intermediate part (t_I) of the keyphrase, the model can learn a multi-token keyphrase. As an example, given a sentence with five tokens (t_1, t_2, t_3, t_4, t_5) of which two (t_2, t_3) are part of a keyphrase, the label would be represented as $(t_O, t_B, t_I, t_O, t_O)$. Figure 2 illustrates our model architecture with the various layers.

4 Experiments

Datasets. We ran our experiment on 2 publicly available keyphrase datasets: PubMed (Gero and Ho, 2019) and INSPEC (Hulth, 2003). PubMed consists of 2532 articles from PubMed Central Open Access Subset with at least 5 author-provided keyphrases while INSPEC contains 200 abstracts of scientific journal papers from Computer Science collected between the years 1998 and 2002. Each document in INSPEC has two sets of keywords assigned: the controlled keywords, which are manually controlled assigned keywords that appear in the Inspec thesaurus but may not appear in the document, and the uncontrolled keywords which are freely assigned by the editors. The union of both sets is considered as the ground-truth in this work. Summary statistics for the datasets are shown in Table 1.

Since we use a sequence labeling formulation of the keyphrase extraction problem, the abstract/keyphrases data pairs are prepared such that each document is a sequence of word tokens, each with positive labels if it occurs in a keyphrase (k_B, k_I), or with a negative label (k_O).

Experiment Settings. As baseline models, we train BiLSTM and BiLSTM-CRF with 100-dimension Glove pre-trained embedding vectors (Pennington et al., 2014). We also train BiLSTM-CRF with two 768-dimension contextual embeddings, BERT (Devlin et al., 2019) and BioBERT (Lee et al., 2020). DAKE (Santosh et al., 2020), a state-of-the art baseline, uses a sentence enrich-

Table 2: Model performance on different datasets

Model	PubMed	INSPEC
BiLSTM (GloVe)	0.543	0.427
BiLSTM-CRF (GloVe)	0.554	0.453
BiLSTM-CRF (BERT)	0.604	0.581
BiLSTM-CRF (BioBERT)	0.622	0.464
DAKE	0.623	0.463
Ours	0.644	0.586

ing process from all the documents using sentence embedding. To replicate their work, we used the BERT model to extract sentence embeddings for each document and enrich the representation. Finally, our model is trained using BERT word embeddings for the INSPEC dataset and BioBERT embeddings for the PubMed dataset.

The results reported are from three runs using 80/20/20 split for train/val/test sets respectively. The BiLSTM, and BiLSTM-CRF are optimized during training using stochastic gradient descent with the learning rate 0.0001. Gradient clipping and drop-out are used to prevent overflow and overfitting. We select the model with the best F1 score on the validation set over three runs. The final test scores reported are the averages running the best model on the test sets.

The code was implemented in Tensorflow 2.4.1 and the code is available at <https://github.com/ZHgero/keyphrases centrality.git>.

5 Results

The performance comparisons between the baselines and our model are shown in Table 2. Our model performs significantly better on the PubMed dataset compared to the existing baselines. In particular, the results show the impact of the centrality layer as it provides a boost in AUC of 0.02 from BiLSTM-CRF (BioBERT) to our model. The improvement gained from our model is not as large on the INSPEC dataset. We hypothesize that for the centrality constraint to be effective, the input sequence should be relatively longer. The sentences in the INSPEC dataset are much shorter hence the difficulty in learning the central theme.

We also compared our models with several state-of-the-art unsupervised approaches including SingleRank (Litvak and Last, 2008), PositionRank (Florescu and Caragea, 2017), TopicRank (Bougouin et al., 2013), and SIFRank (Sun et al.,

Table 3: Ranking comparison on the PubMed dataset

Model	F1@5	F1@10	F1@15
SingleRank	15.2	16.3	19.2
PositionRank	18.3	18.3	20.9
TopicRank	26.4	28.7	29.2
SIFRank	32.3	48.4	56.2
Ours	34.8	53.1	62.6

2020). Table 3 presents the comparison on the PubMed dataset. Since the unsupervised methods are ranking-based methods, the performances are evaluated in terms of F1-measure when a fixed number of keyphrases are extracted. To convert our model into a ranking model, we compute the probability for the predicted keyphrases by using an independence assumption after calculating the marginal probabilities from the CRF layer. The results illustrate that our model outperforms previous unsupervised methods by a significant margin.

In Figure 3, we compare keyphrases tagged by the BioBERT model and our model on a sample abstract. The true positives are colored blue while false negatives are in red. We observe that the BioBERT model fails to identify ‘chronic thromboembolic pulmonary hypertension’ as an important keyphrase whereas our model correctly identifies it. This may be due to the single occurrence of ‘pulmonary hypertension’ in the input text. Meanwhile our model leverages the document embedding to ‘understand’ that pulmonary hypertension is semantically relevant in the context of the entire abstract. We also observe a similar pattern with the keyphrase ‘duration of anticoagulation’. Even though both models fail to capture the entire phrase, our model identifies ‘anticoagulation’ as a strong candidate because of its semantic meaning in the context of the whole abstract.

The figure also illustrates the limitation of the models as both struggle with common words such as ‘post’ and ‘high’ that are attached as prefixes to important keywords. ‘High risk’, ‘duration of’ and ‘post-’ are considered unimportant by both models. This can be explained by the fact that such words usually occur outside a keyphrase boundary and get overlooked even when they appear with important words. False positives by both models are important terms as the phrases are very relevant in the context of abstract but were not selected by the authors.

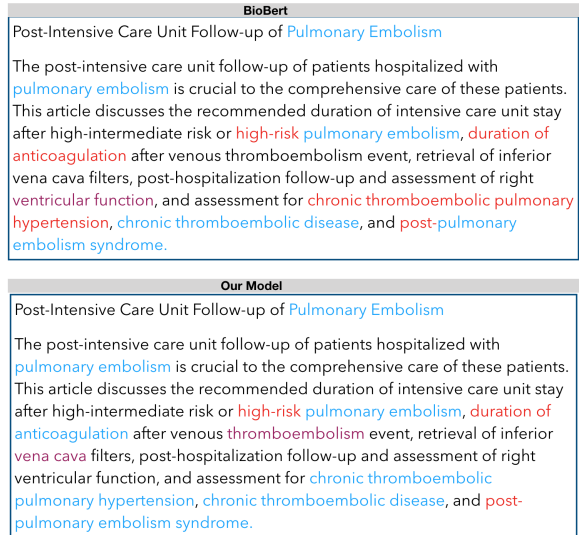


Figure 3: Comparison of keyphrases tagged by two models. True positives are colored blue while false negatives are in red. Purple represents keys that are false positive.

6 Conclusion

In this paper, we proposed a keyphrase extraction method that focuses on identifying words which are central to the document semantics. The problem of keyphrase extraction is posed as a sequence labeling task where each token is tagged as either a keyphrase or not. In addition to our novel centrality constraint layer, we have used Bi-LSTM layers to capture the long term dependencies among the input sequences. Finally, we have a CRF layer which is well suited to capture the dependencies from the output labels. Empirical results on two datasets show that our method gains significant improvement in the PubMed dataset while performing slightly better on the INSPEC dataset.

Acknowledgements

This work is supported by funding from the National Institute of Health awards #5K01LM012924 and #1R01LM013323.

References

- Pubmed search engine. <https://pubmed.ncbi.nlm.nih.gov/>. Accessed: 2021-03-20.
- Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. **Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents**. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2551–2557.
- Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *conference of the canadian society for computational studies of intelligence*, pages 40–52. Springer.
- Santosh Kumar Bharti and Korra Sathya Babu. 2017. Automatic keyword extraction for text summarization: A survey. *arXiv preprint arXiv:1704.03242*.
- Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. **TopicRank: Graph-based topic ranking for keyphrase extraction**. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551.
- Frans Coenen, Paul Leng, Robert Sanderson, and Yanbo J Wang. 2007. Statistical identification of key phrases for text classification. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 838–853. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **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, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Corina Florescu and Cornelia Caragea. 2017. **Position-Rank: An unsupervised approach to keyphrase extraction from scholarly documents**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115.
- Zelalem Gero and Joyce C Ho. 2019. Namedkeys: Unsupervised keyphrase extraction for biomedical documents. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 328–337.
- Sujatha Das Gollapalli and Cornelia Caragea. 2014. **Extracting keyphrases from research papers using citation networks**. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1629–1635.
- Sujatha Das Gollapalli, Xiaoli Li, and Peng Yang. 2017. **Incorporating expert knowledge into keyphrase extraction**. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3180–3187.
- Maria P. Grineva, Maxim N. Grinev, and Dmitry Lizorkin. 2009. **Extracting key terms from noisy and multitheme documents**. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 661–670.
- Khaled M Hammouda, Diego N Matute, and Mohamed S Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *International workshop on machine learning and data mining in pattern recognition*, pages 265–274. Springer.
- Anette Hulth. 2003. **Improved automatic keyword extraction given more linguistic knowledge**. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.
- Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 756–757.
- Steve Jones and Mark S Staveley. 1999. Phrasier: a system for interactive document retrieval using keyphrases. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167.
- Su Nam Kim and Min-Yen Kan. 2009. **Re-examining automatic keyphrase extraction approaches in scientific articles**. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009)*, pages 9–16.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Marina Litvak and Mark Last. 2008. **Graph-based keyword extraction for single-document summarization**. In *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, pages 17–24.
- Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009a. **Unsupervised approaches for automatic keyword extraction using meeting transcripts**. In *Proceedings of Human Language Technologies: The*

- 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 620–628.
- Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. [Clustering to find exemplar terms for keyphrase extraction](#). In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266.
- Naw Naw and Ei Ei Hlaing. 2013. Relevant words extraction method for recommendation system. *Bulletin of Electrical Engineering and Informatics*, 2(3):169–176.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *European Conference on Information Retrieval*, pages 328–335. Springer.
- Tokala Yaswanth Sri Sai Santosh, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. Dake: Document-level attention for keyphrase extraction. In *European Conference on Information Retrieval*, pages 392–401. Springer.
- Kamal Sarkar. 2014. A keyphrase-based approach to text summarization for english and bengali documents. *International Journal of Technology Diffusion (IJTD)*, 5(2):28–38.
- Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang, and Chaoran Zhang. 2020. Sifrank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, 8:10896–10906.
- Nedelina Teneva and Weiwei Cheng. 2017. [Saliencerank: Efficient keyphrase extraction with topic modeling](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 530–535.
- Stamatina Thomaidou and Michalis Vazirgiannis. 2011. [Multiword keyword recommendation system for online advertising](#). In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2011, Kaohsiung, Taiwan, 25-27 July 2011*, pages 423–427.
- Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information retrieval*, 2(4):303–336.
- Peter D Turney. 2002. Learning to extract keyphrases from text. *arXiv preprint cs/0212013*.
- Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.
- Rui Wang, Wei Liu, and Chris McDonald. 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software Engineering Research Conference*, volume 39, pages 1–8.
- Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pages 129–152.
- Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. 2006. [Finding advertising keywords on web pages](#). In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 213–222.
- Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. [Keyphrase extraction using deep recurrent neural networks on Twitter](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845.