Improving Embedding-based Unsupervised Keyphrase Extraction by Incorporating Structural Information

Mingyang Song, Huafeng Liu*, Yi Feng, Liping Jing*
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, Beijing, China
mingyang.song@bjtu.edu.cn

Abstract

Keyphrase extraction aims to extract a set of keyphrases with the central idea of the document. In a structured document, there are certain locations (e.g., the title or the first sentence) where a keyphrase is most likely to appear. However, when extracting keyphrases from the document, most existing embedding-based unsupervised keyphrase extraction models ignore the indicative role of the highlights in certain locations, leading to wrong keyphrases extraction. In this paper, we propose a new Highlight-Guided Unsupervised Keyphrase Extraction model (HGUKE) to address the above issue. Specifically, HGUKE first models the phrase-document relevance via the highlights of the documents. Next, HGUKE calculates the cross-phrase relevance between all candidate phrases. Finally, HGUKE aggregates the above two relevance as the importance score of each candidate to rank and extract keyphrases. The experimental results on three benchmarks demonstrate that HGUKE outperforms the state-of-the-art unsupervised keyphrase extraction baselines.

1 Introduction

Keyphrase extraction is the fundamental task of automatically extracting a set of salient phrases from a document that concisely describes its primary content (Hasan and Ng, 2014; Song et al., 2023a). Figure 1 shows an example of the source document and its corresponding keyphrases.

Recent developments in pre-trained language models (Devlin et al., 2019) have heightened the need for utilizing pre-trained embeddings on natural language processing tasks, which significantly improves the performance of embedding-based unsupervised keyphrase extraction models (Sun et al., 2020; Liang et al., 2021; Zhang et al., 2022). Existing embedding-based models mainly consist of two components: candidate keyphrase extraction and keyphrase importance estimation (Hasan and Ng, 2014; Song et al., 2021, 2022a). The former extracts continuous words from the document as candidate keyphrases through heuristic rules, and the latter estimates the importance of candidate phrases by matching similarity with their corresponding document.

Title:
Measuring keyboard response delays by comparing keyboard and joystick inputs
Abstract:
The response characteristics of PC keyboards have to be identified when they are used as response devices in psychological experiments. In the past, the proposed method has been to check the characteristics independently by means of external measurement equipment. However, with the availability of different PC models and the rapid pace of model change, there is an urgent need for the development of convenient and accurate methods of checking. The method proposed consists of raising the precision of the PC's clock to the microsecond level and using a joystick connected to the MIDI terminal of a sound board to give the PC an independent timing function. Statistical processing of the data provided by this method makes it possible to estimate accurately the keyboard scanning interval time and the average keyboard delay time…
Keyphrases:
keyboard response delay measurement, joystick input, keyboard input, pc keyboard, psychological experiment, model change, check, pc clock precision, midi terminal, sound board, independent timing function, statistical data process, keyboard scan interval time, average keyboard delay time

Figure 1: Randomly sampled document with its corresponding keyphrases from the benchmark keyphrase extraction dataset Inspec. Bold orange represents the content related to the title, and underlined indicates the content related to the first sentence.

Generally, the source document has both salient information and noises (redundant content). Hence, there may be a deviation when directly using the phrase-document relevance as the importance score of each candidate to select keyphrases. For many specific-domain documents (e.g., news or scientific articles), the highlights (the title or the first sentence) typically contains the central information of the source document (as shown in Figure 1), which has more significant guidance for extracting keyphrases. However, the recent embedding-based unsupervised keyphrase extraction models ignore the effect of the highlight information, leading to extract wrong keyphrases.

Motivated by the above issues, we propose a new Highlight-Guided Unsupervised Keyphrase Extraction model (HGUKE), which estimates the impor-

*Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2023, pages 1041–1048

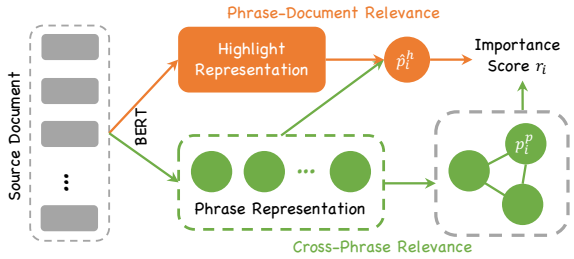

Figure 2: The model architecture of HGUKE.

tance score of each candidate phrase by jointly considering the global and local relevance between candidate phrases and their corresponding document. Concretely, HGUKE first calculates the global relevance by leveraging the highlights rather than the whole document and then locally computes the cross-phrase relevance between all candidate keyphrases, as illustrated in Figure 2. Finally, HGUKE aggregates the global and local relevance as the importance score of each candidate keyphrase to rank and extract keyphrases. Experimental results demonstrate that the proposed model HGUKE outperforms the recent state-of-the-art embedding-based unsupervised keyphrase extraction baselines on three benchmark keyphrase extraction datasets.

2 Methodology

2.1 Candidate Keyphrase Extraction

To extract candidate keyphrases from the source document, we follow the previous studies (Liang et al., 2021; Song et al., 2022b; Ding and Luo, 2021) and leverage Stanford CoreNLP Tools[1] for tokenizing, part-of-speech tagging and noun phrase chunking. Concretely, in our model, the regular expression $\{< NN.*|JJ > * < NN.* >\}$ is designed to extract noun phrases as the candidate keyphrases via the python package NLTK[2].

2.2 Phrase and Document Encoding

After constructing a set of candidate keyphrases $\mathbf{P} = \{p_1, ..., p_i, ..., p_{|\mathbf{P}|}\}$ for the source document via the above method, we adopt the pre-trained language model BERT (Devlin et al., 2019) as the embedding layer to obtain pre-trained word embeddings $\mathbf{H} = \{h_1, ..., h_m, ..., h_{|\mathbf{D}|}\}$ for the source document $\mathbf{D} = \{w_1, ..., w_m, ..., w_{|\mathbf{D}|}\}$ where h_m indicates the m-th word in the document.

[1]https://stanfordnlp.github.io/CoreNLP/
[2]https://github.com/nltk

Next, we leverage word embeddings to obtain candidate keyphrase representations. To capture the central semantics of the candidate keyphrases, we obtain candidate keyphrase representations by leveraging the max pooling operation, which is a simple and effective parameter-free approach and can be calculated as follows,

$$h_{p_i} = \text{Max-Pooling}(\{h_1, ..., h_k, ..., h_{|p_i|}\}), \quad (1)$$

where h_{p_i} is the representation of the i-th candidate keyphrase and $|p_i|$ indicates the length of p_i. Specifically, h_k represents the word in the document associated with the candidate keyphrase p_i. At the same time, we use the mean pooling operation to obtain the highlight representation h_s of the document.

2.3 Phrase-Document Relevance

To obtain more relevant candidates, we model the similarity between candidate phrases and the corresponding document as follows,

$$p_i^h = \frac{1}{||h_s - h_{p_i}||_1}, \quad (2)$$

where p_i^h denotes the phrase-document relevance of i-th candidate keyphrases and $||\cdot||_1$ indicates the Manhattan Distance.

For news and scientific articles, keyphrases often appear at the beginning or front position (Florescu and Caragea, 2017a,b), which means that the position information is important and indicative for extracting keyphrases. For example, the word appearing at 2-th, 5-th and 10-th, has a weight $\rho_i = 1/2 + 1/5 + 1/10 = 0.8$. Inspired by the previous work (Florescu and Caragea, 2017b; Liang et al., 2021), we adopt a position regularization as follows, $\rho_i = \text{softmax}(e^{1/i})$, where ρ_i is the position regularization factor of the i-th candidate phrase. Then, the weighted phrase-document relevance $\hat{p}_i^h$ can be re-calculated as follows,

$$\hat{p}_i^h = p_i^h \cdot \rho_i, \quad (3)$$

Here, we finally employ $\hat{p}_i^h$ to estimate the phrase-document relevance of the i-th candidate phrase.

2.4 Cross-Phrase Relevance

Generally, the phrase-document relevance is calculated between the highlight information and each candidate independently, and consequently, it cannot determine which candidates are better than the

Model	DUC2001			Inspec			SemEval2010		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
Statistical Keyphrase Extraction Models									
TF-IDF (Jones, 2004)	9.21	10.63	11.06	11.28	13.88	13.83	2.81	3.48	3.91
YAKE (Campos et al., 2018)	12.27	14.37	14.76	18.08	19.62	20.11	11.76	14.4	15.19
Graph-based Keyphrase Extraction Models									
TextRank (Mihalcea and Tarau, 2004)	11.80	18.28	20.22	27.04	25.08	36.65	3.80	5.38	7.65
SingleRank (Wan and Xiao, 2008)	20.43	25.59	25.70	27.79	34.46	36.05	5.90	9.02	10.58
TopicRank (Bougouin et al., 2013)	21.56	23.12	20.87	25.38	28.46	29.49	12.12	12.90	13.54
PositionRank (Florescu and Caragea, 2017b)	23.35	28.57	28.60	28.12	32.87	33.32	9.84	13.34	14.33
MultipartiteRank (Boudin, 2018)	23.20	25.00	25.24	25.96	29.57	30.85	12.13	13.79	14.92
Embedding-based Keyphrase Extraction Models									
EmbedRankd2v (Bennani-Smires et al., 2018)	24.02	28.12	28.82	31.51	37.94	37.96	3.02	5.08	7.23
KeyGames (Saxena et al., 2020)	24.42	28.28	29.77	32.12	40.48	40.94	11.93	14.35	14.62
SIFRank (Sun et al., 2020)	24.27	27.43	27.86	29.11	38.80	39.59	-	-	-
JointGL (Liang et al., 2021)	28.62	35.52	36.29	32.61	40.17	41.09	13.02	19.35	21.72
MDERank (Zhang et al., 2022)	23.31	26.65	26.42	27.85	34.36	36.40	13.05	18.27	20.35
HGUKE	**31.31**	**37.24**	**38.31**	**34.18**	**41.05**	**42.16**	**14.07**	**20.52**	**23.10**

Table 1: Performance of the selected baselines and our model on DUC2001, Inspec and SemEval2010 test sets. F1 scores on the top 5, 10, and 15 keyphrases are reported. The best results are bolded in the table.

others. To determine which candidate phrases are more salient than the others, we sum the semantic relatedness between the i-th candidate phrases and all candidates as the cross-phrase relevance. Thus, it calculates the local relevance as follows,

$$p_i^p = \sum_{j=1, j\neq i} (h_{p_i} h_{p_j}^\top - \lambda \delta_i). \quad (4)$$

where $\delta_i = \text{Mean}(\sum_{j=1,j\neq i} h_{p_i} h_{p_j}^\top)$. Here, we treat δ_i as a de-noisy factor to filter the noises, which is far different from the i-th candidate keyphrase in the document.

2.5 Relevance Aggregation

We aggregate the phrase-document relevance and the cross-phrase relevance into a whole score as the importance score of each candidate via a simple multiplication,

$$r_i = \hat{p}_i^h \cdot p_i^p \quad (5)$$

where r_i indicates the importance score of the i-th candidate phrase. Then, we rank all candidates with their importance score r_i and extract top-ranked k phrases as keyphrases of the source document.

3 Experiments and Results

3.1 Experimental Settings

This paper conducts experiments on three benchmark and popular used keyphrase datasets, which includes DUC2001 (Wan and Xiao, 2008), Inspec (Hulth, 2003), and SemEval2010 (Kim et al., 2010). Due to page limits, please refer to the corresponding articles for the details of the three datasets.

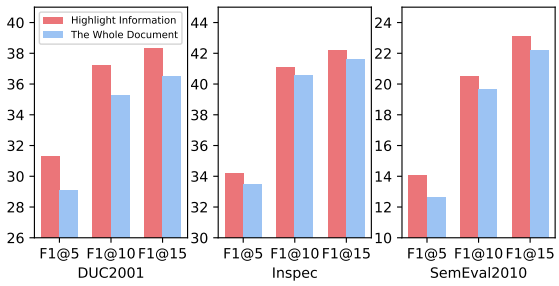

Figure 3: The results of calculating the phrase-document relevance via the whole document and the highlights.

Following the previous work (Liang et al., 2021; Ding and Luo, 2021; Song et al., 2023b), we use the standard practice and evaluate the performance of our model in terms of f-measure at the top-K keyphrases (F1@K) and adopt stemming to both extracted keyphrases and gold truth. Concretely, we report F1@5, F1@10, and F1@15 of each model on three benchmark datasets.

We adopt the pre-trained language model BERT (Devlin et al., 2019) as the backbone of our model, initialized from their pre-trained weights. In our experiments, λ is set to 0.9 for three benchmark datasets.

3.2 Overall Performance

Table 1 shows the performance of baselines and our model on three benchmark datasets (DUC2001, In-

Pooling Methods	DUC2001			Inspec			SemEval2010		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
Max-Pooling	25.43	33.24	36.10	33.95	**41.21**	42.12	9.92	17.20	21.54
Mean-Pooling	**31.31**	**37.24**	**38.31**	**34.18**	41.05	**42.16**	**14.07**	**20.52**	**23.10**

Table 2: The results of different pooling methods for document embedding.

Different Similarity Measures	DUC2001			Inspec			SemEval2010		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
Cosine Similarity	30.26	36.14	37.19	33.70	40.51	41.53	13.15	20.78	23.26
Euclidean Distance	30.67	36.65	37.95	34.04	41.02	**42.24**	13.43	20.18	**23.44**
Manhattan Distance	**31.31**	**37.24**	**38.31**	**34.18**	**41.05**	42.16	**14.07**	**20.52**	23.10

Table 3: The results of different similarity measure methods for the phrase-document relevance.

spec, and SemEval2010). The results show that our method significantly improves over state-of-the-art unsupervised keyphrase extraction baselines. Compared with the current state-of-the-art models, our model achieves significantly better performance on F1@5, F1@10, and F1@15 evaluation metrics, demonstrating the effectiveness of estimating the importance of candidate phrases by leveraging the highlights to calculate the relevance.

Compared with EmbedRank (Bennani-Smires et al., 2018), KeyGames (Saxena et al., 2020), and SIFRank (Sun et al., 2020), HGUKE achieves significant improvement, which benefits from using the highlights to calculate the importance score of each candidate keyphrase. Compared with the best baseline JointGL, our model achieves better performance on several benchmark keyphrase extraction datasets in all evaluation metrics. The main reason for this improvement is that we use the highlights as the guidance information instead of the whole document when estimating the importance of keyphrases.

3.3 Ablation Test

The ablation experiments on three benchmark keyphrase extraction datasets are shown in Figure 3. It can be seen from the results that using the highlight information can significantly improve the performance of keyphrase extraction, which benefits from estimating the importance score of each candidate by using its corresponding highlight information rather than the whole document. We consider the main reason is that the title or the first sentence of the document usually has a strong guidance for extracting keyphrases.

3.4 Impact of Pooling Methods

In this section, we study different pooling methods, including mean- and max-pooling operations. For all pooling methods, HGUKE using the last BERT layer achieves the best results, demonstrating that HGUKE benefits from stronger contextualized semantic representations. We can see the results in Table 2 that the document encoded via the mean-pooling operation obtains the best performance.

3.5 Impact of Different Similarity Measures

Our model adopts Manhattan Distance to measure the textual similarity between candidate phrases and the highlight information. Furthermore, we attempt to employ different measures to estimate the phrase-document relevance. The results of different similarity measures are shown in Table 3, and we can see that the advantage of Manhattan Distance is obvious.

4 Related Work

Most existing unsupervised keyphrase extraction methods can be mainly divided into four categories: statistics-based, topic-based, graph-based, and embedding-based models. Specifically, statistics-based models (Salton and Buckley, 1988; Witten et al., 1999) usually extract keyphrases by estimating the importance of candidate phrases with different statistic features, such as word frequency feature, phrase position feature, linguistic features of natural language, etc. Topic-based models (Liu et al., 2009, 2010) typically utilize topic information to determine whether a candidate phrase is a keyphrase. Graph-based models (Mihalcea and Tarau, 2004; Grineva et al., 2009) represent the

document as a graph and rank candidate phrases by graph-based similarities.

Embedding-based models usually adopt the pre-trained embeddings to obtain document and candidate phrase representations and calculate the importance score of each candidate depending on the obtained representations. Benefiting from the development of transformer-based pre-trained language models (Devlin et al., 2019) in the natural language processing field, embedding-based models (Bennani-Smires et al., 2018; Sun et al., 2020; Liang et al., 2021) have achieved outstanding performance. Concretely, embedding-based models mainly consist of two procedures: candidate keyphrase representation and keyphrase importance estimation (Hasan and Ng, 2014; Song et al., 2023a). The first procedure utilizes natural language linguistic features to construct candidate keyphrases and represents them by pre-trained embedding approaches (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)). The second procedure estimates the importance of candidate phrases from different perspectives to determine whether a candidate phrase is a keyphrase.

Unlike the existing unsupervised keyphrase extraction models, we use the highlight information of the document to calculate the phrase-document relevance instead the whole document.

5 Conclusion and Future Work

In this paper, we incorporate structural information to improve the performance of embedding-based unsupervised keyphrase extraction. Specifically, in this paper, we propose a new Highlight-Guided Unsupervised Keyphrase Extraction model (HGUKE), which calculates the phrase-document relevance via the highlight information instead of the whole document to select relevant candidate phrases. Extensive experiments demonstrate that HGUKE outperforms the state-of-the-art unsupervised baselines. Future research may investigate adopting different structural information of the source document to improve the performance of unsupervised keyphrase extraction.

6 Acknowledgments

We thank the three anonymous reviewers for carefully reading our paper and their insightful comments and suggestions. This work was partly supported by the Fundamental Research Funds for the Central Universities (2019JBZ110); the National Natural Science Foundation of China under Grant 62176020; the National Key Research and Development Program (2020AAA0106800); the Beijing Natural Science Foundation under Grant L211016; CAAI-Huawei MindSpore Open Fund; and Chinese Academy of Sciences (OEIP-O-202004).

7 Limitations

There are still some limitations of our work. In the future, we plan to enhance the procedure of extracting candidate keyphrase, to improve the upper bound of the performance of keyphrase extraction. One possible way is to generate candidate phrases of the document by utilizing the high-level semantic relatedness (e.g., attention weights) instead of using the surface-or syntactic-level information.

References

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *CoNLL*, pages 221–229. Association for Computational Linguistics.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *NAACL-HLT (2)*, pages 667–672. Association for Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *IJCNLP*, pages 543–551. Asian Federation of Natural Language Processing / ACL.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 806–810. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Haoran Ding and Xiao Luo. 2021. Attentionrank: Unsupervised keyphrase extraction using self and cross attentions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1928.

Corina Florescu and Cornelia Caragea. 2017a. A position-biased pagerank algorithm for keyphrase extraction. In *AAAI*, pages 4923–4924. AAAI Press.

Corina Florescu and Cornelia Caragea. 2017b. Positionrank: An unsupervised approach to keyphrase

extraction from scholarly documents. In *ACL (1)*, pages 1105–1115. Association for Computational Linguistics.

Maria P. Grineva, Maxim N. Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *WWW*, pages 661–670. ACM.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *ACL (1)*, pages 1262–1273. The Association for Computer Linguistics.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP*.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *SemEval@ACL*, pages 21–26. The Association for Computer Linguistics.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Unsupervised keyphrase extraction by jointly modeling local and global context. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 155–164, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *HLT-NAACL*, pages 620–628. The Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *CoRR*, abs/1907.11692.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *EMNLP*, pages 366–376. ACL.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*, pages 404–411. ACL.

Gerard Salton and Chris Buckley. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523. Also available in Sparck Jones and Willett (1997).

Arnav Saxena, Mudit Mangal, and Goonjan Jain. 2020. Keygames: A game theoretic approach to automatic keyphrase extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2037–2048.

Mingyang Song, Yi Feng, and Liping Jing. 2022a. Hyperbolic relevance matching for neural keyphrase extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5710–5720. Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2022b. Utilizing BERT intermediate layers for unsupervised keyphrase extraction. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 277–281, Trento, Italy. Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2023a. A survey on recent advances in keyphrase extraction from pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2153–2164, Dubrovnik, Croatia. Association for Computational Linguistics.

Mingyang Song, Liping Jing, and Lin Xiao. 2021. Importance Estimation from Multiple Perspectives for Keyphrase Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mingyang Song, Lin Xiao, and Liping Jing. 2023b. Learning to extract from multiple perspectives for neural keyphrase extraction. *Computer Speech & Language*, 81:101502.

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.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, pages 855–860. AAAI Press.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255. ACM.

Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, Shiliang Zhang, Bing Li, Wei Wang, and Xin Cao. 2022. Mderank: A masked document embedding rank approach for unsupervised keyphrase extraction. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 396–409. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
3

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
3

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.