A Preliminary Empirical Study on Prompt-based Unsupervised Keyphrase Extraction

Mingyang Song, Yi Feng, Liping Jing

Beijing Key Lab of Traffic Data Analysis and Mining Beijing Jiaotong University, Beijing, China mingyang.song@bjtu.edu.cn

Abstract

Pre-trained large language models can perform natural language processing downstream tasks by conditioning on human-designed prompts. However, a prompt-based approach often requires "prompt engineering" to design different prompts, primarily hand-crafted through laborious trial and error, requiring human intervention and expertise. It is a challenging problem when constructing a prompt-based keyphrase extraction method. Therefore, we investigate and study the effectiveness of different prompts on the keyphrase extraction task to verify the impact of the cherry-picked prompts on the performance of extracting keyphrases. Extensive experimental results on six benchmark keyphrase extraction datasets and different pretrained large language models demonstrate that (1) designing complex prompts may not necessarily be more effective than designing simple prompts; (2) individual keyword changes in the designed prompts can affect the overall performance; (3) designing complex prompts achieve better performance than designing simple prompts when facing long documents.

1 Introduction

Keyphrase extraction aims at automatically extracting a set of phrases from the input document to summarize its core topics and primary information (Hasan and Ng, 2014; Song et al., 2023b). Generally, keyphrase extraction models are trained on many document-keyphrase data pairs (Sun et al., 2021; Song et al., 2021, 2023h, 2022a). These models demonstrate exceptional extractive capabilities for obtaining keyphrases from the given document, especially for Large Language Model (LLM) based keyphrase extraction systems. However, the quality of keyphrases extracted by prompt-based keyphrase extraction models is subject to the quality of the input prompts, whether in unsupervised or supervised settings. Designing proper prompts for keyphrase extraction models based on large pre-trained lan-

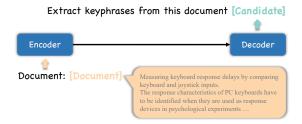


Figure 1: The illustration of a prompt-based keyphrase extraction model under an encoder-decoder architecture.

guage models is challenging (Wu et al., 2022; Song et al., 2023e; Kong et al., 2023).

In natural language processing, prompt-based learning is a new paradigm to replace fine-tuning large pre-trained language models on downstream tasks (Liu et al., 2023). Different from fine-tuning, prompt, the form of natural language, is more consistent with the pre-training task of models. Prompt-based learning has been widely used in many natural language processing tasks. In this paper, we analyze different prompts for unsupervised keyphrase extraction, leveraging the capability of large pre-trained language models with an encoder-decoder architecture.

As presented in Figure 1, the general process of extracting keyphrases uses an encoder-decoderbased large pre-trained language model. It is necessary to design appropriate prompts to assist the model in outputting keyphrases for the input document, which means the design of prompts directly affects the performance of the prompt-based keyphrase extraction models. Typically, prompts for effectively extracting keyphrases are predominantly hand-crafted through laborious trial and error, requiring human intervention and expertise (Kong et al., 2023; Song et al., 2023c,e). However, previous studies on keyphrase extraction has not systematically experimented with and analyzed whether complex or simple prompts might be more effective.

In this paper, we directly leverage a large pretrained language model with an encoder-decoder architecture (i.e., T5 (Raffel et al., 2020)) to measure the similarity without fine-tuning. Specifically, after extracting keyphrase candidates from the original document, we feed the input document into the encoder and calculate the probability of generating the candidate with a designed prompt by the decoder. The higher the probability, the more important the candidate. Experimental results on six benchmark keyphrase extraction datasets and different models demonstrate that (1) designing a complex prompt may not necessarily be more effective than designing a simple prompt; (2) individual keyword changes in a designed prompt can affect the overall performance; (3) designing a complex prompt achieve better performance than designing a simple prompt when facing long documents.

2 Related Work

Generally, unsupervised keyphrase extraction methods are divided into three categories: statistics-, graph-, and embedding-based models. Statistics-based models (Jones, 2004; Campos et al., 2018) estimate the importance score of each candidate keyphrase by utilizing their statistical characteristics such as frequency, position, capitalization, and other features that capture the context information. The graph-based models (Mihalcea and Tarau, 2004; Bougouin et al., 2013; Boudin, 2018) are first proposed by TextRank (Mihalcea and Tarau, 2004), which treats each candidate keyphrase as a vertice, constructs edges according to the co-occurrence of candidates, and determines the weight of vertices through the PageRank algorithm.

Embedding-based models (Saxena et al., 2020; Sun et al., 2020; Bennani-Smires et al., 2018; Song et al., 2022c, 2023d; Zhang et al., 2022) have achieved SOTA performance, benefiting from the recent development of the pre-trained language models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). However, these algorithms perform poorly on long texts due to the length mismatch between the document and the candidate. (Zhang et al., 2022) solves the problem by replacing the embedding of the candidate with that of the masked document but fails to utilize the PLMs without fine-tuning fully. To address such issues, (Kong et al., 2023) utilizes prompt-based learning for unsupervised keyphrase extraction.

In this paper, different from the existing mod-

els, we investigate the meaning of the design of prompts in the unsupervised keyphrase extraction task, leveraging the capability of pre-trained language models with an encoder-decoder architecture, such as T5 (Raffel et al., 2020).

3 Methodology

The main pipeline of prompting large language models for unsupervised keyphrase extraction is illustrated in Figure 1. Following the recent work, we extract candidates from the document via heuristic rules. After obtaining candidates, we first incorporate the document into a designed prompt as the input of the encoder and then calculate the probability of generating the candidate as the importance score with a designed prompt by the decoder. Finally, the importance score is used to rank and extract keyphrases. In analyzing the impact of different prompts in this paper, no additional parameter designs were introduced for fairness.

3.1 Candidate Extraction

In this paper, we follow the previous studies and leverage the common practice (Song et al., 2023f; Zhang et al., 2022) to extract candidate keyphrases using the regular expression < NN.* | JJ > * < NN.* > after tokenization and POS tagging.

3.2 Importance Estimation

Precisely, we fill the encoder template with the original input document and fill the decoder template with one candidate at a time. Then, we obtain the sequence probability $p(y_i|y_{< i})$ of the decoder template with the candidate based on pre-trained language models, such as T5 (Raffel et al., 2020). The length-normalized log-likelihood has been widely used due to its superior performance (Brown et al., 2020). Hence, we calculate the probability for one candidate as follows:

$$\pi_c = -\frac{1}{l_c} \sum_{i=m}^{m+l_c-1} \log p(y_i|y_{< i}).$$
 (1)

where l_c is the length of each candidate keyphrase. Here, we use π_c , whose value is positive, to evaluate the importance of candidates in ascending order. Then, select the top K candidate keyphrases with the highest scores as the final set of keyphrases.

4 Experiment

We present the used datasets and evaluation metrics, the implementation details, and the results.

Index	Encoder	Decoder	Model	F1@K		
		Decoder		5	10	15
	"[Document]"	"[Candidate]"	T5-BASE	11.46	16.62	18.68
p_1			T5-3B	11.36	16.52	18.83
			FLAN-T5	11.56	16.80	19.19
	Article: "[Document]"	This article mainly talks about "[Candidate]"	T5-BASE	15.76	21.74	23.35
p_2			T5-3B	20.98	25.82	26.26
			FLAN-T5	18.46	23.37	24.20
	Article: "[Document]"	Keyphrases of this article are "[Candidate]"	T5-BASE	12.74	18.10	19.89
p_3			T5-3B	14.46	19.72	21.57
			FLAN-T5	14.29	19.23	20.68

Table 1: The results of the same prompt with different keywords. "[Document]" is filled with the document, and "[Candidate]" is filled with the candidate. F1@K here is the average of six datasets.

Index	Encoder	Decoder	F1@K			
		Decoder	5	10	15	
p_1	"[Document]"	"[Candidate]"	11.46	16.62	18.68	
$p_{1,1}$	Article: "[Document]"	"[Candidate]"	11.41	16.68	18.96	
$p_{1,2}$	"[Document]"	t]" Keyphrases: "[Candidate]"		21.30	23.03	
$p_{1,3}$	Article: "[Document]"	Keyphrases: "[Candidate]"	16.42	21.39	22.88	
p_2	Article: "[Document]"	cument]" This article mainly talks about "[Candidate]"		21.74	23.35	
$p_{2,1}$	Passage: "[Document]"	This passage mainly talks about "[Candidate]"	15.55	21.30	23.04	
$p_{2,2}$	Book: "[Document]" This book mainly talks about "[Candidate]"		16.25	21.88	23.45	
$p_{2,3}$	Document: "[Document]"	ocument: "[Document]" This document mainly talks about "[Candidate]"		21.65	23.36	
$p_{2,4}$	Paper: "[Document]"	This paper mainly talks about "[Candidate]"	16.06	21.66	23.28	
$p_{2,5}$	Content: "[Document]"	This content mainly talks about "[Candidate]"		21.54	23.28	
$p_{2,6}$	Text: "[Document]"	This text mainly talks about "[Candidate]"	16.34	21.87	23.49	
p_3	Article: "[Document]"	Keyphrases of this article are "[Candidate]"		18.10	19.89	
$p_{3,1}$	Article: "[Document]"	Keywords of this article are "[Candidate]"	13.31	18.39	19.90	
$p_{3,2}$	Article: "[Document]"	The keyphrases of this article are "[Candidate]"	13.61	18.82	20.52	
$p_{3,3}$	Article: "[Document]"	Extract keyphrases from this article: "[Candidate]"	18.14	22.81	23.89	

Table 2: The performance of several prompts with different keywords. "[Document]" is filled with the document, and "[Candidate]" is filled with the candidate. F1@K here is the average of six datasets.

4.1 Datasets

In this paper, we conduct experiments on six widely used keyphrase extraction benchmark datasets, such as Inspec (Hulth, 2003), DUC2001 (Wan and Xiao, 2008), SemEval2010 (Kim et al., 2010), SemEval2017 (Augenstein et al., 2017), Nus (Nguyen and Kan, 2007), and Krapivin (Krapivin and Marchese, 2009).

4.2 Evaluation Metrics

Following the previous researches (Song et al., 2023d,f,i,g; Kong et al., 2023), we adopt F1 on the top 5, 10, and 15 ranked candidates to evaluate the results in this paper. When calculating F1 score, duplicate candidate keyphrases are removed, and stemming is applied.

4.3 Implementation Details

We adopt the pre-trained language model T5 (Raffel et al., 2020) as the backbone, initialized from their pre-trained weights. Among them, there are two versions used in this paper, such as "T5-base" and "T5-3B". Furthermore, we also use the pre-trained language model Flan-T5-base (Chung et al., 2022) as the backbone to conduct experiments. Similar to the recent work, to match the settings of BERT (Devlin et al., 2019), the maximum length for the inputs of the encoder is set to 512. In addition, we utilize the code from Kong et al. (2023) to complete the experiments in this paper. The difference is that we do not introduce any adjustable parameters. For more details, please refer to Kong et al. (2023).

F1@K	Model	Dataset					Avia	
		Inspec	SemEval2017	SemEval2010	DUC2001	NUS	Krapivin	Avg.
F1@5	T5-base $(p_{1,3})$	27.17	22.37	10.56	18.23	11.43	8.74	16.4
	T5-base $(p_{2,6})$	29.90	24.50	13.25	21.91	11.49	11.82	18.8
	T5-base $(p_{3,3})$	27.71	22.99	13.15	20.37	13.14	11.50	18.1
	T5-3B (p _{1,3})	28.79	21.65	10.96	16.43	9.73	9.80	16.2
	T5-3B $(p_{2,6})$	29.55	23.88	16.44	22.11	17.01	17.08	21.0
	T5-3B $(p_{3,3})$	28.04	22.52	11.56	18.23	9.84	9.51	16.6
	FLAN-T5 $(p_{1,3})$	28.28	21.77	10.76	12.05	9.50	9.35	15.2
	FLAN-T5 $(p_{2,6})$	28.66	23.94	14.25	17.18	12.86	12.91	18.3
	FLAN-T5 $(p_{3,3})$	28.54	22.17	11.46	12.60	9.56	9.43	15.6
	T5-base $(p_{1,3})$	33.65	31.19	15.56	22.79	14.31	10.83	21.3
	T5-base $(p_{2,6})$	35.45	33.82	18.03	25.64	15.01	13.42	23.5
	T5-base $(p_{3,3})$	33.75	31.82	17.63	23.87	16.45	13.37	22.8
	T5-3B $(p_{1,3})$	35.15	31.55	17.15	22.21	14.13	11.35	21.9
F1@10	T5-3B $(p_{2,6})$	34.58	33.72	19.39	26.86	20.26	17.37	25.3
	T5-3B $(p_{3,3})$	34.60	32.40	16.91	23.33	14.22	11.02	22.0
	FLAN-T5 $(p_{1,3})$	34.52	30.91	17.15	16.73	13.48	10.94	20.6
	FLAN-T5 $(p_{2,6})$	34.02	32.95	18.91	21.20	16.63	14.17	22.9
	FLAN-T5 $(p_{3,3})$	34.85	31.02	17.07	17.96	13.56	10.83	20.8
	T5-base $(p_{1,3})$	33.95	34.85	17.69	23.89	15.61	11.30	22.8
	T5-base $(p_{2,6})$	34.82	37.33	18.82	26.15	16.32	13.37	24.4
F1@ 15	T5-base $(p_{3,3})$	33.79	35.54	18.82	24.91	17.31	13.00	23.8
	T5-3B $(p_{1,3})$	35.38	35.10	18.96	23.35	16.35	11.74	23.4
	T5-3B $(p_{2,6})$	34.51	36.54	20.42	27.54	20.76	15.93	25.9
	T5-3B $(p_{3,3})$	35.08	35.88	18.96	24.71	15.39	11.68	23.6
	FLAN-T5 $(p_{1,3})$	35.02	34.69	19.42	19.35	15.14	11.21	22.4
	FLAN-T5 $(p_{2,6})$	33.93	36.10	20.09	22.84	17.67	13.14	23.9
	FLAN-T5 $(p_{3,3})$	34.90	34.61	19.62	19.72	15.36	10.56	22.4

Table 3: The performance of keyphrase extraction on six datasets. The best results are highlighted in bold.

4.4 Results

As mentioned before, we mainly focus on investigating and studying the effectiveness of different prompts on the keyphrase extraction task to verify the impact of the cherry-picked prompts on the performance of extracting keyphrases in this paper. Therefore, we design three types of prompts (ranging from simple to complex) suitable for extracting keywords. Then, we conduct experiments on different large pre-trained language models, further replace keywords in prompts, and analyze the necessity of cherry-picked prompts. All results are displayed in Table 1, Table 2, and Table 3. Next, we analyze the experimental results in detail.

From the results in Table 1, it can be seen that when the prompt (p_1) is not used at all, the performance of T5-base and T5-3b are both poor, and even the performance of T5-3b is not as good as

T5-base, while Flan-T5 achieved the best effect. After using more detailed prompts (p_2 and p_3), it can be found that the results of T5-3B and Flan-T5 are significantly better than those of the T5-base. Furthermore, it can be seen from Table 1 that using p_2 as a prompt can achieve better performance than using p_3 as a prompt, whether using T5-base, T5-3B, or Flan-T5.

Many existing approaches attempt to construct various prompts, such as modifying different keywords in the prompts, to achieve better performance. Hence, we also analyzed the impact of different keywords in the modification prompt on the results. Taking inspiration from existing methods (Kong et al., 2023; Song et al., 2023c,e), we modified the keywords in the three prompts used in this paper and verified their performance. The results are shown in Table 2. From the results, we can find that the designed prompts $(p_{1,3}, p_{2,6}, p_{3,3})$

obtain the best results, respectively. However, we found that changing different keywords has little effect on the results in most cases, indirectly indicating the effectiveness of constructing refined prompts but requiring a lot of experimentation.

The results in Table 3 show that T5-3B performs significantly better than T5-base and Flan-T5 on long document datasets, such as the SemEval2010 dataset. Meanwhile, the results of $p_{2,6}$ are considerably better than those of $p_{1,3}$ and $p_{3,3}$, which indicates the necessity of designing a complex prompt. The difference in the results is insignificant with different prompts, so a refined prompt design is not a reasonable strategy based on existing results. On the contrary, automatic prompt generation or search should be more convenient and efficient.

5 Conclusion

In this paper, we investigate the effectiveness of different prompts to verify the impact of the cherrypicked prompts on the performance of extracting keyphrases. Extensive experimental results on six benchmark keyphrase extraction datasets and different pre-trained large language models demonstrate that (1) designing complex prompts may not necessarily be more effective than designing simple prompts in most cases; (2) individual keyword changes in prompts affect the overall performance; (3) designing complex prompts achieve better performance than designing simple prompts when facing long documents. Future research may be possible to better extend similar ideas from phrase-level to sentence-level information extraction (i.e., the extractive summarization task (Song et al., 2022b, 2023a)) in the future. In addition, it might be possible to construct a new long-context benchmark, such as the needle-in-a-haystack¹ or the Counting-Stars (Song et al., 2024), through the keyphrase extraction task.

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. In SemEval@ACL, pages 546–555. Association for Computational Linguistics.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

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*.

¹https://github.com/gkamradt/LLMTest_ NeedleInAHaystack

- 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.
- Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xiaoyan Bai. 2023. Promptrank: Unsupervised keyphrase extraction using prompt. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 9788–9801. Association for Computational Linguistics.
- M. Krapivin and M. Marchese. 2009. Large dataset for keyphrase extraction.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.
- 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.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*, pages 404–411. ACL.
- Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *ICADL*, volume 4822 of *Lecture Notes in Computer Science*, pages 317–326. Springer.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.
- 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. A preliminary exploration of extractive multi-document summarization in hyperbolic space. In *Proceedings*

- of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022, pages 4505–4509. ACM.
- Mingyang Song, Yi Feng, and Liping Jing. 2022c. Utilizing BERT intermediate layers for unsupervised keyphrase extraction. In 5th International Conference on Natural Language and Speech Processing, ICNLSP 2022, Trento, Italy, December 16-17, 2022, pages 277–281. Association for Computational Linguistics.
- Mingyang Song, Yi Feng, and Liping Jing. 2023a. Hisum: Hyperbolic interaction model for extractive multi-document summarization. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 4 May 2023*, pages 1427–1436. ACM.
- Mingyang Song, Yi Feng, and Liping Jing. 2023b. A survey on recent advances in keyphrase extraction from pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2108–2119. Association for Computational Linguistics.
- Mingyang Song, Xuelian Geng, Songfang Yao, Shilong Lu, Yi Feng, and Liping Jing. 2023c. Large language models as zero-shot keyphrase extractors: A preliminary empirical study. *CoRR*, abs/2312.15156.
- Mingyang Song, Haiyun Jiang, Lemao Liu, Shuming Shi, and Liping Jing. 2023d. Unsupervised keyphrase extraction by learning neural keyphrase set function. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2482–2494. Association for Computational Linguistics.
- Mingyang Song, Haiyun Jiang, Shuming Shi, Songfang Yao, Shilong Lu, Yi Feng, Huafeng Liu, and Liping Jing. 2023e. Is chatgpt A good keyphrase generator? A preliminary study. *CoRR*, abs/2303.13001.
- 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*, pages 2726–2736. Association for Computational Linguistics.
- Mingyang Song, Huafeng Liu, Yi Feng, and Liping Jing. 2023f. Improving embedding-based unsupervised keyphrase extraction by incorporating structural information. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1041–1048. Association for Computational Linguistics.
- Mingyang Song, Huafeng Liu, and Liping Jing. 2023g. HyperRank: Hyperbolic ranking model for unsupervised keyphrase extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16070–16080, Singapore. Association for Computational Linguistics.

- Mingyang Song, Lin Xiao, and Liping Jing. 2023h. Learning to extract from multiple perspectives for neural keyphrase extraction. *Comput. Speech Lang.*, 81:101502.
- Mingyang Song, Pengyu Xu, Yi Feng, Huafeng Liu, and Liping Jing. 2023i. Mitigating over-generation for unsupervised keyphrase extraction with heterogeneous centrality detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16349–16359, Singapore. Association for Computational Linguistics.
- Mingyang Song, Mao Zheng, and Xuan Luo. 2024. Counting-stars: A multi-evidence, position-aware, and scalable benchmark for evaluating long-context large language models.
- Si Sun, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, and Jie Bao. 2021. Capturing global informativeness in open domain keyphrase extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 275–287. Springer.
- 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.
- Huanqin Wu, Baijiaxin Ma, Wei Liu, Tao Chen, and Dan Nie. 2022. Fast and constrained absent keyphrase generation by prompt-based learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 March 1, 2022*, pages 11495–11503. AAAI Press.
- 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*, pages 396–409, Dublin, Ireland. Association for Computational Linguistics.