Is ChatGPT A Good Keyphrase Generator? A Preliminary Study

Mingyang Song[♠], Haiyun Jiang[♣*], Shuming Shi[♣]
Songfang Yao[♠], Shilong Lu[♠], Yi Feng[♠], Huafeng Liu[♠], Liping Jing[♠*]
[♣]Tencent AI Lab, Shenzhen, China
[♠]Beijing Key Lab of Traffic Data Analysis and Mining
[♠]Beijing Jiaotong University, Beijing, China
mingyang.song@bjtu.edu.cn

Abstract

The emergence of ChatGPT has recently garnered significant attention from the computational linguistics community. To demonstrate its capabilities as a keyphrase generator, we conduct a preliminary evaluation of ChatGPT for the keyphrase generation task. We evaluate its performance in various aspects, including keyphrase generation prompts, keyphrase generation diversity, multi-domain keyphrase generation, and long document understanding. Our evaluation is based on six benchmark datasets, and we adopt the prompt suggested by OpenAI while extending it to six candidate prompts. We find that ChatGPT performs exceptionally well on all six candidate prompts, with minor performance differences observed across the datasets. Based on our findings, we conclude that ChatGPT has great potential for keyphrase generation. Moreover, we discover that ChatGPT still faces challenges when it comes to generating absent keyphrases. Meanwhile, in the final section, we also present some limitations and future expansions of this report. We have released the used data[1].

1 Introduction

ChatGPT[2] is an intelligent chatbot developed by OpenAI based on the InstructGPT (Ouyang et al., 2022). ChatGPT aims to provide excellent conversation responses based on the prompts provided. OpenAI highlights that ChatGPT can provide detailed responses, answer follow-up questions, and challenge incorrect or inappropriate prompts while maintaining a natural dialogue format. Furthermore, ChatGPT encompasses a wide range of natural language processing abilities, such as code debugging, question answering, article paraphrasing, keyphrase generation, and logic reasoning, to name a few. We are particularly intrigued by the capability of keyphrase generation of ChatGPT, as we would like to understand how well it performs compared to the state-of-the-art keyphrase generation models. We focus on narrowing the gap between ChatGPT and such keyphrase generation models and exploring new and innovative ways to improve the performance of keyphrase generation and enhance its features to provide even better results. Note that we consider that keyphrases and keywords are different in this report. Keyphrases are often phrases of two or more words rather than single words (keywords).

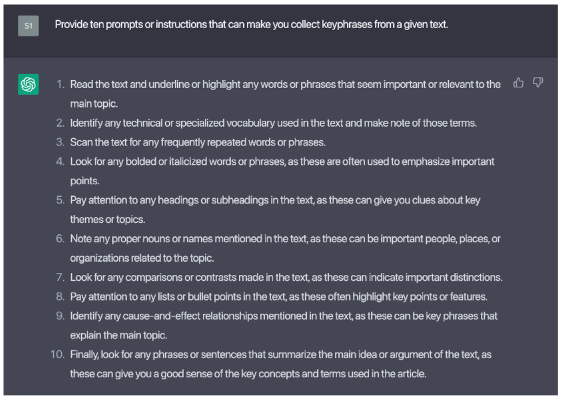

Figure 1: Prompts or instructions advised by ChatGPT for keyphrase generation (Date: 2023.03.01).

In this report, we summarize a preliminary study of ChatGPT on keyphrase generation to gain a better understanding of it. Specifically, we focus on five perspectives:

• **Keyphrase Generation Prompts**:

ChatGPT is essentially a large language model which needs prompts as guidance to trigger its ability to generate keyphrases. The different style of prompts or instructions may affect the quality of generated keyphrases. Generally, the performance of ChatGPT to obtain keyphrases depends largely on the prompts. A good prompt often obtains better performance than a bad one. Therefore, we verify

*Corresponding Author

[1] https://github.com/MySong7NLPer/ChatGPT_as_Keyphrase_Generator

[2] https://openai.com/blog/chatgpt

Test Set	Domain	Type	# Doc.	Avg. # Words	Present KPs (%)
KP20k (Meng et al., 2017)	Scientific Abstract	Short	**20,000**	179.8	57.40
Inspec (Hulth, 2003)	Scientific Abstract	Short	**500**	128.7	55.69
Nus (Nguyen and Kan, 2007)	Scientific Abstract	Short	**211**	219.1	67.75
Krapivin (Krapivin and Marchese, 2009)	Scientific Abstract	Short	**400**	182.6	44.74
SemEval2010 (Kim et al., 2010)	Scientific Abstract	Short	**100**	234.8	42.01
OpenKP (Xiong et al., 2019)	Open Web Domain	Long	**6,616**	900.4	100.00

Table 1: Information of adopted test sets. **# Doc.** is the number of documents in the dataset. **Avg. # Words** is the average number of words for documents. **Present KPs (%)** indicates the percentage of keyphrases, which are presented in the documents. **Note that this report uses all of the test data rather than sampling part from it**.

	Prompts
Tp1[†]	**`Extract keywords from this text:`** `[Document]`
Tp2	`Generate keywords from this text: [Document]`
Tp3	`Extract keyphrases from this text: [Document]`
Tp4	`Generate keyphrases from this text: [Document]`
Tp5	`Generate present and absent keywords from this text: [Document]`
Tp6	`Generate present and absent keyphrases from this text: [Document]`

Table 2: Six prompts are designed for chatting with ChatGPT to collect keywords or keyphrases from the given text document. † indicates the prompt provided by OpenAI[2].

the quality of six prompts to verify the performance of ChatGPT in the task of keyphrase generation.

• **Keyphrase Generation Diversity**:

Keyphrase generation aims to generate a set of phrases that can *cover the main topics discussed in a given document* (Hasan and Ng, 2014). Recent advances in keyphrase generation have made remarkable progress, demonstrated through improved quality metrics such as F1-score. However, the importance of diversity in keyphrase generation has been largely ignored (Bahuleyan and Asri, 2020). Following previous studies (Ye et al., 2021; Xie et al., 2022), we report the average numbers of unique present and absent keyphrases and the average duplication ratios of all predicted keyphrases to investigate the ability of ChatGPT to generate diverse keyphrases.

• **Multi-domain Keyphrase Generation**:

ChatGPT is a versatile language model that performs various natural language processing tasks across multiple domains. We are interested in evaluating how well ChatGPT performs on different domains, considering differences in text styles (e.g., news documents vs. scientific articles). This will give us a better understanding of its strengths and limitations in different contexts.

• **Different Length Document Understanding**:

Different lengths of documents usually correspond to different optimal processing strategies, especially for long documents. For example, in the task of keyphrase generation, while it may be possible to design better algorithms to handle a large number of candidates in long documents, we believe that employing sophisticated features, especially those that encode background knowledge, will enable keyphrases and non-keyphrases to be distinguished more efficiently, even in the presence of a large number of candidates. At the same time, how to encode long documents is also a problem worth studying. Therefore, we tested the ability of ChatGPT as a general large language model to handle documents of different lengths.

2 ChatGPT for Keyphrase Generation

2.1 Evaluation Setting

We briefly introduce the evaluation setting, which mainly includes the compared baselines, datasets, and evaluation metrics. Note that each time a new query is made to ChatGPT, we clear conversations to avoid the influence of previous samples, which is similar to Bang et al. (2023).

2.1.1 Baselines

We compare ChatGPT with several state-of-the-art keyphrase generation systems: catSeq (Meng et al., 2017), catSeqTG-2RF1 (Chan et al., 2019), ExHiRD-h (Chen et al., 2020), SetTrans (Ye et al., 2021), and WR-SetTrans (Xie et al., 2022). Furthermore, we also compare ChatGPT

[2]https://platform.openai.com/examples/default-keywords

Model	KP20k		Inspec		Nus		Krapivin		SemEval	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
RNN-based Models										
catSeq (Meng et al., 2017)	0.291	0.367	0.225	0.262	0.323	0.397	0.269	0.354	0.242	0.283
catSeqTG-2RF1 (Chan et al., 2019)	0.321	0.386	0.253	0.301	0.375	0.433	0.300	0.369	0.287	0.329
ExHiRD-h (Chen et al., 2020)	0.311	0.374	0.253	0.291	N/A	N/A	0.286	0.347	0.284	0.335
Transformer-based Models										
SetTrans (Ye et al., 2021)	0.358	0.392	0.285	0.324	0.406	0.450	0.326	**0.364**	0.331	0.357
WR-SetTrans (Xie et al., 2022)	**0.370**	**0.378**	0.330	0.351	**0.428**	**0.452**	**0.360**	0.362	**0.360**	**0.370**
PLM-based Models										
SciBART-base‡ (124M)	0.341	0.396	0.275	0.328	0.373	0.421	0.282	0.329	0.270	0.304
BART-base‡ (140M)	0.322	0.388	0.270	0.323	0.366	0.424	0.270	0.336	0.271	0.321
T5-base‡ (223M)	0.336	0.388	0.288	0.339	0.388	0.440	0.302	0.350	0.295	0.326
ChatGPT (gpt-3.5-turbo)										
ChatGPT w/ **Tp1**†	0.186	0.160	0.298	0.417	0.319	0.225	0.239	0.187	0.267	0.216
ChatGPT w/ **Tp2**	0.180	0.149	0.310	0.433	0.314	0.239	0.243	0.197	0.275	0.240
ChatGPT w/ **Tp3**	0.161	0.141	0.383	0.463	0.281	0.211	0.218	0.184	0.268	0.214
ChatGPT w/ **Tp4**	0.160	0.136	0.322	0.393	0.208	0.187	0.170	0.163	0.233	0.212
ChatGPT w/ **Tp5**	0.174	0.161	0.330	0.427	0.336	0.261	0.244	0.204	0.281	0.243
ChatGPT w/ **Tp6**	0.179	0.165	**0.401**	**0.476**	0.287	0.253	0.230	0.213	0.262	0.233

Table 3: Results of present keyphrase generation on five datasets. F1 scores on the top 5 and M keyphrases are reported, where M is a variable cut-off equal to the number of predictions. The results of the baselines are reported in their corresponding papers. Specifically, ‡ indicates the results are reported in Wu et al. (2022). The best results are highlighted in bold.

with several state-of-the-art keyphrase extraction systems: JointKPE (BERT-base) (Sun et al., 2021), JointKPE (RoBERTa-base) (Sun et al., 2021), HyperMatch (Song et al., 2022), and KIEMP (Song et al., 2021). By default, the results in this report come from the ChatGPT version on 2023.03.01. For new results, we will mark the updated version information correspondingly.

2.1.2 Datasets

We evaluate ChatGPT and all the baselines on five test datasets: Inspec (Hulth, 2003), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin and Marchese, 2009), SemEval (Kim et al., 2010), and KP20k (Meng et al., 2017). As implemented in (Ye et al., 2021; Xie et al., 2022), we perform data preprocessing, including tokenization, lowercasing, replacing all digits with the symbol ⟨digit⟩, and removing duplicated instances. Note that all the baselines were trained by using the KP20k training set. Furthermore, we verify the keyphrase generation ability of ChatGPT in real-world scenarios on the OpenKP dataset (Xiong et al., 2019) where *documents are from diverse domains* and have variant content quality. Table 1 summarizes the information of the used dataset.

2.1.3 Evaluation Metrics

Following previous studies (Meng et al., 2017; Chan et al., 2019; Ye et al., 2021; Xie et al., 2022), we adopt macro averaged F1@5 and F1@M to evaluate the quality of both present and absent keyphrases. When using F1@5, blank keyphrases are added to make the keyphrase number reach five if the prediction number is less than five. Mainly, we employ the Porter Stemmer to remove the identical stemmed keyphrases.

2.2 Keyphrase Generation Prompts

To design the prompts or instructions for triggering the keyphrase generation ability of ChatGPT, we seek inspiration from ChatGPT by asking it for advice. Concretely, we ask ChatGPT with the following instruction:

- `Provide ten prompts or instructions that can make you collect keyphrases from a given text.`

and obtain the results in Figure 1. The generated prompts seem reasonable, as they all involve extracting some core features of the keyphrases. However, we found that using these prompts to ask ChatGPT, what we got was not a list of keyphrases but something like a summary. Fortunately, OpenAI provides an official prompt on how to extract keywords. Based on this instruction, we extend it to six different candidate prompts, as shown in Table 2. Specifically, we expect to analyze ChatGPT's understanding of several of the following through these six prompts:

Model	KP20k		Inspec		Nus		Krapivin		SemEval	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
RNN-based Models										
catSeq (Meng et al., 2017)	0.015	0.032	0.004	0.008	0.016	0.028	0.018	0.036	0.016	0.028
catSeqTG-2RF1 (Chan et al., 2019)	0.027	0.050	0.012	0.021	0.019	0.031	0.030	0.053	0.021	0.030
ExHiRD-h (Chen et al., 2020)	0.016	0.032	0.011	0.022	N/A	N/A	0.022	0.043	0.017	0.025
Transformer-based Models										
SetTrans (Ye et al., 2021)	0.036	0.058	0.021	0.034	0.042	0.060	0.047	0.073	0.026	0.034
WR-SetTrans (Xie et al., 2022)	**0.050**	**0.064**	0.025	0.034	**0.057**	**0.071**	**0.057**	**0.074**	**0.040**	**0.043**
PLM-based Models										
SciBART-base‡ (124M)	0.029	0.052	0.016	0.028	0.033	0.053	0.033	0.054	0.018	0.022
BART-base‡ (140M)	0.022	0.042	0.010	0.017	0.026	0.042	0.028	0.049	0.016	0.021
T5-base‡ (223M)	0.017	0.034	0.011	0.020	0.027	0.051	0.023	0.043	0.014	0.020
ChatGPT (gpt-3.5-turbo)										
ChatGPT w/ **Tp1**†	0.045	0.053	0.016	0.030	0.001	0.001	0.003	0.004	0.006	0.007
ChatGPT w/ **Tp2**	0.045	0.052	0.025	0.042	0.008	0.009	0.007	0.011	0.005	0.007
ChatGPT w/ **Tp3**	0.041	0.045	0.027	0.047	0.003	0.004	0.004	0.008	0.002	0.002
ChatGPT w/ **Tp4**	0.038	0.039	**0.030**	**0.041**	0.009	0.012	0.011	0.015	0.004	0.005
ChatGPT w/ **Tp5**	0.036	0.026	0.017	0.015	0.004	0.005	0.006	0.005	0.004	0.004
ChatGPT w/ **Tp6**	0.041	0.025	0.029	0.024	0.009	0.009	0.005	0.009	0.005	0.006

Table 4: Results of absent keyphrase generation on five datasets. F1 scores on the top 5 and M keyphrases are reported, where M is a variable cut-off equal to the number of predictions. The results of the baselines are reported in their corresponding papers. Specifically, ‡ indicates the results are reported in Wu et al. (2022). The best results are highlighted in bold.

1. *What is the difference between "keyword" and "keyphrase" ?*

   From the results in Table 3, it seems that ChatGPT does not have a gap in the understanding between "keyword" and "keyphrase".

2. *What is the difference between "extract" and "generate" ?*

   As you can see from Table 5, the prompts using "generate" (e.g., **Tp**2 and **Tp**4) predict more absent keyphrases than the prompts using "extract" (e.g., **Tp**1† and **Tp**3) when chatting with ChatGPT. This shows that ChatGPT makes a distinction between present and absent keyphrases.

3. *What is the difference between "present keyphrase" and "absent keyphrase" ?*

   Figure 2 illustrates the answer of ChatGPT to this question. From the results, we find that ChatGPT distinguishes between present and absent keyphrases. When explicitly asked in the instruction to generate present and absent will significantly increase the number of generated absent keyphrases by ChatGPT. More importantly, we find that ChatGPT is really not good at generating absent keyphrases in this report.

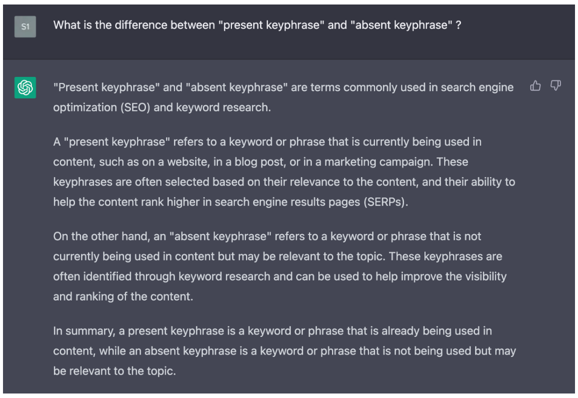

Figure 2: The answer of ChatGPT about the difference between "present keyphrase" and "absent keyphrase".

2.3 Multi-domain Keyphrase Generation

To verify the adaptability of ChatGPT under different domain documents, we tested it on the OpenKP dataset. The results show that ChatGPT is highly adaptable to multi-domain data and performs better than the state-of-the-art keyphrase extraction baselines.

2.4 Keyphrase Generation Diversity

To investigate the ability of ChatGPT to generate diverse keyphrases, we measure the average numbers of unique present and absent keyphrases and the average duplication ratio of all the predicted keyphrases following recent studies (Ye et al., 2021;

Model	KP20K			SemEval		
	#PK	#AK	Dup	#PK	#AK	Dup
Oracle	**3.31**	**1.95**	**0.000**	**6.12**	**8.31**	**0.000**
SetTrans	**5.10**	**2.01**	0.080	4.62	2.18	0.080
WR-SetTrans	6.35	3.26	0.100	**5.94**	**3.60**	0.100
ChatGPT w/ **Tp1**†	14.14	5.58	**0.005**	24.53	1.06	**0.009**
ChatGPT w/ **Tp2**	16.29	6.33	0.006	21.78	1.66	0.013
ChatGPT w/ **Tp3**	13.02	6.99	0.024	22.48	2.07	0.009
ChatGPT w/ **Tp4**	13.29	7.92	**0.005**	11.97	3.65	0.006
ChatGPT w/ **Tp5**	12.00	14.36	0.027	20.48	7.74	0.020
ChatGPT w/ **Tp6**	9.68	13.09	0.279	17.24	8.40	0.040

Table 5: Number and duplication ratio of predicted keyphrases on three datasets. "#PK" and "#AK" are the average number of unique present and absent keyphrases respectively. "Dup" refers to the average duplication ratio of predicted keyphrases. "Oracle" refers to the gold average keyphrase number.

Xie et al., 2022). The results are reported in Table 4. Based on the results, ChatGPT generates more unique keyphrases than all baselines by a large margin and achieves a significantly lower duplication ratio without additional hyper-parameter settings.

2.5 Long Document Understanding

Handling long documents is challenging in many natural language processing tasks, often with complex contexts in long documents. Similarly, effectively understanding long documents is an urgent challenge in keyphrase generation. Most of the existing keyphrase extraction baselines (Sun et al., 2021; Song et al., 2021, 2022) need to truncate the input documents to require the input limitations of the used backbone (e.g., BERT (Devlin et al., 2019)), which leads to a large amount of information loss. Meanwhile, extracting keyphrases that are salient to the document meanings is an essential step to semantic document understanding. Therefore, we tested the effect of ChatGPT on understanding long documents, as shown in Table 6. From the results, we find that ChatGTP can understand long documents well. Later we will choose some longer documents (e.g., a document contains approximately 4096 words) as test documents.

3 Conclusion

This report presents a preliminary study of ChatGPT for the keyphrase generation task, including keyphrase generation prompts, keyphrase generation diversity, multi-domain keyphrase generation, and long document understanding. The results

Model	OpenKP	
	F1@3	F1@5
PLM-based Models		
JointKPE (BERT-Base) (Sun et al., 2021)	0.376	0.325
JointKPE (RoBERTa-Base) (Sun et al., 2021)	0.391	0.338
HyperMatch (Song et al., 2022)	**0.394**	0.338
KIEMP (Song et al., 2021)	0.392	0.340
ChatGPT (gpt-3.5-turbo)		
ChatGPT w/ **Tp3**	0.287	**0.436**

Table 6: Results of present keyphrase generation on the OpenKP dataset. F1@3 is the main evaluation metric for this dataset (Xiong et al., 2019). The results of the baselines are reported in their corresponding papers. The best results are highlighted in bold.

from the Inspec dataset in Table 3 and Table 4 show that ChatGPT has a strong keyphrase generation capability. It just needs to design better prompts to guide it. Therefore, in the following section, we list several aspects not considered in this report and argue that these may affect the keyphrase generation performance of ChatGPT.

4 Limitations

We admit that this report is far from complete with various aspects to make it more reliable.

4.1 Powerful Prompt Designing

Generally, when chatting with ChatGPT, the design of prompts can largely influence the results it gives. In this report, we make some improvements based on the prompts given by OpenAI, but they are not necessarily optimal. Therefore, designing more appropriate prompts is the key to effectively exploiting the performance of ChatGPT on the task of keyphrase generation.

4.2 Hyper-Parameter Settings

In realistic scenes, users may not care about setting hyper-parameters of ChatGPT when chatting with ChatGPT. Meanwhile, the settings of the hyper-parameters typically require prior knowledge about ChatGPT. Therefore, we do not consider setting different hyper-parameters in this report, which affected the keyphrase generation performance of ChatGPT. Next, we will further consider the corresponding hyper-parameters of ChatGPT to verify its performance on the keyphrase generation task and give a more detailed analysis.

4.3 Few-Shot Prompting

With the increasing power of large language models, in-context learning has become a new paradigm for natural language processing (Dong et al., 2022), where large language models make predictions only based on contexts augmented with a few examples. It has been a new trend to explore in-context learning to evaluate and extrapolate the ability of large language models (e.g., the emergent ability (Wei et al., 2022)). In this report, we do not consider using some strategies to explore the ability of ChatGPT. In the future direction, we believe it is possible to enhance the performance of large language models in the keyphrase generation task through similar methods.

4.4 Evaluation Metric

Previous studies (Meng et al., 2017; Ye et al., 2021) have mainly used extensions of standard F1-based metrics to measure the performance of keyphrase generation models. Such evaluation metrics usually operate based on exact matches between predicted and gold keyphrases. Such a strategy cannot account for partial matches or semantic similarity. For example, if the prediction is "keyphrase generation model" and the gold is "keyphrase generation system", despite both semantic similarity and partial matching, the score will be 0. These minor deviations are ubiquitous in keyphrase generation yet harshly penalized by the "exact match" evaluation metrics. Therefore, a semantic-based evaluation metric may be more suitable to measure the performance of ChatGPT on the keyphrase generation task. Furthermore, human evaluation can provide more insights for comparing ChatGPT with keyphrase generation baselines.

References

Hareesh Bahuleyan and Layla El Asri. 2020. Diverse keyphrase generation with neural unlikelihood training. *CoRR*, abs/2010.07665.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *CoRR*, abs/2302.04023.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural keyphrase generation via reinforcement learning with adaptive rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy. Association for Computational Linguistics.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1105, Online. Association for Computational Linguistics.

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.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

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*.

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.

M. Krapivin and M. Marchese. 2009. Large dataset for keyphrase extraction.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *ACL*, pages 582–592. Association for Computational Linguistics.

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.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Mingyang Song, Yi Feng, and Liping Jing. 2022. 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*, pages 5710–5720, Seattle, United States. 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.

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.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *CoRR*, abs/2206.07682.

Di Wu, Wasi Uddin Ahmad, and Kai-Wei Chang. 2022. Pre-trained language models for keyphrase generation: A thorough empirical study.

Binbin Xie, Xiangpeng Wei, Baosong Yang, Huan Lin, Jun Xie, Xiaoli Wang, Min Zhang, and Jinsong Su. 2022. Wr-one2set: Towards well-calibrated keyphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 7283–7293. Association for Computational Linguistics.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *EMNLP/IJCNLP (1)*, pages 5174–5183. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2set: Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4598–4608. Association for Computational Linguistics.