DPPM: Data Pyramid Pre-trained Model for Few-Shot Text Summarization

Anonymous ACL submission

Abstract

Text summarization models are often trained with standard datasets like *CNN/DailyMail* and *BBC XSum*. However, recent studies reveal poor annotation qualities of these datasets. Fine-tuning and evaluating models on such data yield conflicting conclusions between N-gram metrics and human evaluation. On the other hand, annotating a large number of high-quality data is infeasible due to the time-consuming property. To address this challenge, we propose a novel Data Pyramid Pre-trained Model (DPPM) for few-shot text summarization. Data Pyramid includes three parts: extractive data, abstractive data, and user-specific data. The first two types of data serve as the foundation, comprising the majority of the training data, thereby bolstering the DPPM's ability to generate domain-general summaries, and the user-specific data can align our model with human preferences. Experimental results on re-annotated *CNN/DailyMail* and *BBC XSum* datasets show that our DPPM (based on BART-Large) surpasses the performance of 175B GPT-3 and achieves the single-model SOTA.

1 Introduction

Text summarization is a pivotal aspect of natural language processing, aiming to produce a coherent and concise summary of a given text document (Mani and Maybury, 1999; Nenkova and McKeown, 2012; Allahyari et al., 2017). Pre-trained models (Vaswani et al., 2017; Radford et al., 2018; Lewis et al., 2019; Raffel et al., 2020) are often employed for this task. Previous studies have primarily categorized text summarization into two types: extractive (Nallapati et al., 2017; Zhou et al., 2020; Zhong et al., 2020) and abstractive summarization (See et al., 2017; Nallapati et al., 2016; Lewis et al., 2019), and experiments are often conducted on popular news datasets such as *CNN/DailyMail* (Nallapati et al., 2016) and *BBC XSum* (Narayan et al., 2018). However, recent investigations have identified significant issues with the annotation quality of these datasets, such as incoherence and factual hallucination (Maynez et al., 2020; Kang and Hashimoto, 2020; Wang et al., 2023), potentially undermining the correlation between N-gram metrics and human judgments (Zhang et al., 2023b). These findings indicate that high-quality data of text summarization is notably rare.

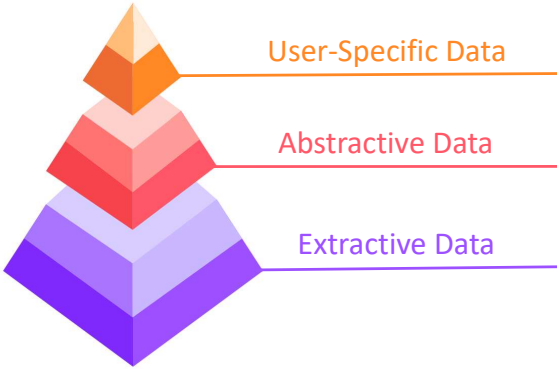

Figure 1: Data pyramid comprises three types of data: extractive data, abstractive data, and user-specific data.

However, it is impractical to annotate a large number of high-quality summary datasets. This infeasibility stems from the fact that annotators often spend twenty or even more minutes reading the original text to generate a summary (Wang et al., 2023). Moreover, ensuring the accuracy and consistency of these summaries typically necessitates cross verification, demanding considerable human and financial resources (Ahuja et al., 2021; Zhang et al., 2023c; Chen et al., 2023). These all result in the scarcity of high-quality data. Therefore, few-shot summarization, which entails training models with few annotated data (Chen and Shuai, 2021), has attracted significant attention within this field (Logan IV et al., 2021; He et al., 2022).

To mitigate the scarcity of high-quality data, we introduce the **D**ata **P**yramid **P**re-trained **M**odel **(DPPM)** for few-shot summarization. The data pyramid includes three parts: extractive data, ab-

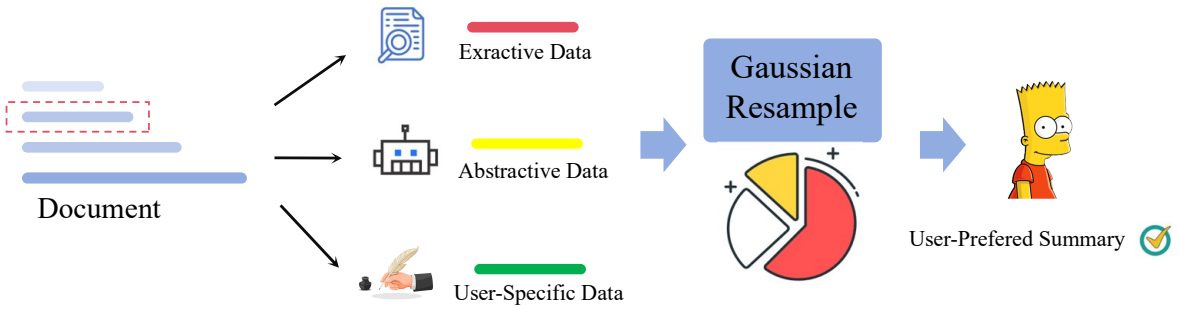

Figure 2: The DPPM training process is illustrated as follows: Initially, extractive data, abstractive data, and user-specific data are acquired through different methods. Then, Gaussian resampling is applied to the first two types of data based on the summary length of user-specific data, to align with user preferences. Finally, training is conducted on the BART model to obtain a pre-trained model that is consistent with user preferences.

stractive data, and user-specific data, as depicted in Figure 1. The volume of these data types diminishes progressively, with the acquisition difficulty increasing correspondingly. Extractive data and abstractive data constitute the foundation of the data pyramid, and fine-tuning on those data can enhance the DPPM's capabilities in domain-general summarization. Most crucially, we focus on user-specific data. In terms of summarization, human preferences significantly diverge, as highlighted by the adage "*there are a thousand Hamlets in a thousand people's eyes*", indicating variability in preferences. Although LLMs demonstrate impressive zero-shot capabilities in summarization tasks (Zhang et al., 2023a; Shi et al., 2023; Goyal et al., 2022), having been pre-trained on extensive web data and fine-tuned through instruct-tuning and reinforcement learning from human feedback (RLHF, Ouyang et al. (2022)), they produce general summaries that may not meet specific user preferences. To ensure that the DPPM-generated summaries more accurately reflect specific user preferences, we propose annotating a modest quantity of user-specific data, thus allowing the model to better grasp and interpret user intentions.

We conducted experiments on the user-reannotated *CNN/DailyMail* and *BBC XSum* datasets (Wang et al., 2023). Our DPPM, based on BART-Large, surpassed the performance of 175B GPT-3, achieving SOTA results as a single model. Additionally, its performance was comparable to the two-stage Summary Chain-of-Thought method proposed by Wang et al. (2023).

2 Approach

Figure 2 illustrates the DPPM training process, which includes two main steps: (i) acquisition of pre-training corpus and (ii) Gaussian resampling. Specifically, we denote the original document as D, comprising n sentences, each represented as d_i. The golden summary is denoted as S.

2.1 Pre-Training Corpus

Extractive Data. Extractive data constitutes the majority of the pre-training corpus and is also the simplest to acquire. We employ adapted Gap Sentence Generation (GSG, Zhang et al. (2020)) to select the most important sentence as the pseudo summary $\hat{S}$:

$$\begin{aligned} r_i &= \text{Rouge}(d_i, D_{\setminus d_i}), \\ \hat{S} &= \text{argmax}_{d_i}\{r_i\}_{i=1}^{n}. \end{aligned} \tag{1}$$

Unlike the method described in Zhang et al. (2020), we extract only a single sentence due to the variability in sentence lengths, rendering control by the number of sentences unreliable. Selection should instead be based on the number of tokens in the Gaussian resample phase.

Abstractive Data. Extractive data aids the DPPM in identifying important sentences within a document, yet it falls short of summarizing crucial information that spans multiple sentences. Meanwhile, LLMs serve as effective zero-shot summarizers for summarization tasks (where the original documents are lengthy, and few-shot learning consumes a significant number of tokens), proficiently extracting summary information across sentences and at the document level. We utilize both system prompts and user prompts to elicit LLMs to summarize the document D and generate the pseudo summary $\hat{S}$, and prompts are related to the specific dataset and can be found in detail in Appendix A.

Dataset	*CNN/DailyMail*	*BBC Xsum*
	Samples / Mean / Std	
Train Data	100/64.46/17.00	100/34.41/9.62
Test Data	100/65.84/15.43	100/32.59/7.90
Extractive Data	229680/54.93/16.97	163232/51.17/53.48
Extractive Data (Re)	80000/54.82/11.67	87848/39.41/6.03
Abstractive Data	57422/102.24/14.89	40809/72.72/19.47
Abstractive Data (Re)	38557/84.15/8.06	26702/40.64/5.04

Table 1: Pseudo summary token lengths statical value (mean and standard deviation) and sample count in data pyramid. "Re" denotes Gaussian resample.

User-Specific Data. User-specific data is the most critical component within the DPPM, utilized for learning user preferences. Training on data generated by adapted GSG and LLMs enables the DPPM to acquire domain-specific knowledge. To generate user-specific summaries, further fine-tuning is necessary on annotated data that contains user preferences. The acquisition of user-specific summary data generally involves human annotation. Unlike the explicit guidance employed by Wang et al. (2023), the DPPM can learn user preferences through data, offering a more universal approach.

2.2 Gaussian Resample

The DPPM draws from three distinct data sources, each with a unique distribution of token lengths for its pseudo summary. Direct training with these disparate distributions can lead to instability in the DPPM training process and the generation of overly long summaries. To mitigate this issue, we introduce Gaussian resampling, modeling the token length distribution of user-specific data as a Gaussian distribution:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2}$$

where μ and σ represent the mean and standard deviation, respectively. Based on μ and σ, the confidence interval for the token length distribution can be determined. With a 95% probability, the confidence interval is $[\mu - 2\sigma, \mu + 2\sigma]$. Using this interval, we resample the content generated by adapted GSG and LLMs, removing samples whose pseudo summaries are excessively long or short.

3 Experimental Setup

3.1 Dataset

We employed two extensively used news datasets, *CNN/DailyMail* and *BBC Xsum*. Due to the comparatively low annotation quality of these datasets, we adopt the element-aware dataset (Wang et al., 2023), annotated according to the "Lasswell Communication Model" (Lasswell, 1948), which we regard as reflecting a type of user-specific preference. To assess the DPPM's efficacy, we randomly split the element-aware dataset into a train set and a test set, each comprising 100 samples. Extractive data and abstractive data are generated from the original*CNN/DailyMail* and *BBC Xsum* datasets' train set, divided according to an 80% and 20% split, respectively, and then generated based on adpated GSG and LLMs.

Table 1 shows the token length distribution of pseudo summary in the data pyramid. Before the Gaussian resample, there was a significant difference in the token distribution between extractive data, abstractive data, and user-specific data. After resampling, the disparity has been noticeably reduced. Furthermore, after resampling, the number of extractive data entries from *CNN/DailyMail* exceeded that of abstractive data. We randomly discarded some, retaining 80,000 samples.

3.2 Model

For generating abstractive data, we employ LLaMA-2-7B (Touvron et al., 2023). For pre-trained models, BART-Large (Lewis et al., 2019) is selected for further fine-tuning. Additionally, we also adopt previous baselines, T5 (Raffel et al., 2020) and PEGASUS (Zhang et al., 2020), for a comprehensive comparison. All PLMs weights are downloaded from Hugging Face.[1] Refer to Appendix B for more details.

4 Results and Analysis

Table 2 shows the overall performance comparison of DPPM, LLMs, and traditional PLMs. We can conclude with two key findings:

- Fine-tuning PLMs with user-specific data significantly enhances performance. In contrast, fine-tuning LLMs with a small amount of user-specific data proves less effective.

- The performance of DPPM surpasses all other models and is marginally inferior to the 175B GPT-3 employing SumCoT (Wang et al., 2023). However, the prompt-based SumCoT aligns closely with the data annotation strategy, limiting its scalability. Conversely, DPPM learns the

[1] https://huggingface.co

Model \ Dataset	*CNN/DailyMail*				*BBC Xsum*			
	ROUGE-1	**ROUGE-2**	**ROUGE-L**	**BERTScore**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**BERTScore**
BART	41.38	18.35	26.05	0.8720	34.86	12.22	24.19	0.8677
T5	42.06	17.18	27.92	0.8763	34.26	11.86	24.62	0.8677
PEGASUS	41.39	15.66	27.26	0.8706	29.26	7.56	21.26	0.8825
LLaMA2-7B	44.78	18.83	29.65	0.8985	37.99	14.20	28.72	0.8952
LLaMA2-7B-FT	44.37	18.17	28.96	0.8906	37.08	14.07	28.57	0.8937
175B GPT-3	42.98	19.48	28.33	0.8943	38.50	15.09	29.09	**0.8981**
175B GPT-3 w/CoT	49.73	26.10	36.29	0.9080	44.36	19.93	34.70	0.9053
BART (+FT)	46.57	21.97	32.00	0.9040	40.19	14.95	28.74	0.8915
BART (+DPPM)	**48.04**	**22.67**	**33.38**	**0.9050**	**41.92**	**17.46**	**30.48**	0.8963

Table 2: Performance comparison of DPPM, LLMs, and Pre-trained models includes fine-tuning BART and LLaMA2-7B with User-Specific Data. GPT-3 results are sourced from SumCoT (Wang et al., 2023). We utilize the widely adopted automatic metrics ROUGE-1/2/L and BERTScore for evaluation.

Few-shots	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
10	44.72	18.96	29.48	0.8855
50	47.38	21.67	31.73	0.8897
100	48.04	22.67	33.38	0.9050

Table 3: The ROUGE-1/2/L and BERTScore for varying few-shots of user-specific data on *CNN/DailyMail*.

summarization model directly from data, offering an end-to-end solution.

Compare to LLMs Fine-tuning. We use LoRA (Hu et al., 2021) to finetune the LLaMA2-7B, despite having significantly fewer trainable parameters compared to BART-Large, the memory consumption during fine-tuning exceeds that of fine-tuning BART-Large, making training unfeasible on consumer-grade hardware, such as the RTX 4090 or 3090. Moreover, fine-tuning with only 100 samples fails to enhance performance and may even lead to a decrease. This is because high-quality fine-tuning typically requires datasets on the order of tens of thousands (Deng et al., 2023; Zhao et al., 2024). This fine-tuning process may also impact the LLMs' other capabilities, such as math reasoning and logic reasoning.

Role of User-Specific Data. Figure 3 displays the results of fine-tuning BART using various methods. As observed, the automatic evaluation performance is poor when fine-tuning without the user-specific data. Simultaneously, Ahuja et al. (2021) proposed a mixed training that mixes all data for training. Nonetheless, experimental results indicate that training on data generated by adapted GSG and LLMs first and then fine-tuning with user-specific data can generate a better summary.

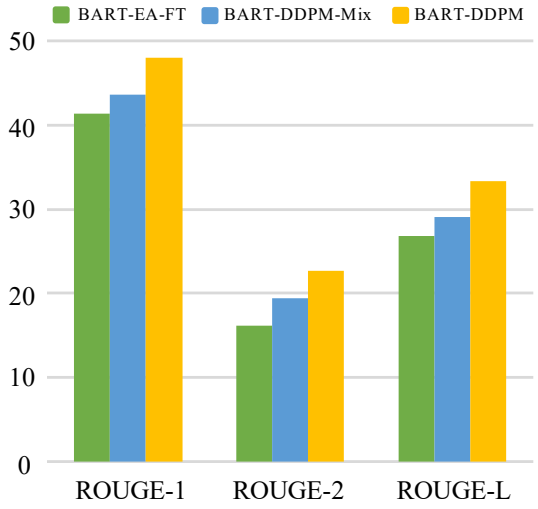

Figure 3: The performance of various training methods on *CNN/DailyMail*. BART-EA-FT: fine-tuning with **E**xtractive and **A**bstractive data. BART-DPPM-Mix: fine-tuning with mixed data pyramid. DPPM: fine-tuning BART-EA-FT with user-specific data.

Few-shots User-Specific Data. Table 3 illustrates the impact of varying amounts of user-specific data on DPPM's ability to learn user summary patterns. It can be observed that with only 10 training samples, DPPM's performance already surpasses that of all pre-trained models and LLMs. Furthermore, as user-specific data increases, the model's performance improves correspondingly.

5 Conclusion

We introduced a novel training paradigm for few-shot text summarization. Through experimental validation, our DPPM, by integrating three different types of data, can learn user-specific summary patterns from the data, effectively mitigating the issue of scarce high-quality training data.

Limitations

Due to the scarcity of high-quality text annotation data, our experiments were conducted solely on the element-aware dataset proposed by Wang et al. (2023). The element-aware annotations for CNN/DailyMail and BBC Xsum are based on news elements, embodying only one type of user annotation pattern. On the other hand, Element-Aware contains only 200 samples, resulting in both the training and test sets comprising merely 100 samples each, indicating a limited train and test size.

Ethics Statement

We utilize publicly available datasets and weight parameters for model training and data generation, all of which are accompanied by bibliographic citations, ensuring no ethical issues are involved.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2021. Aspectnews: Aspect-oriented summarization of news documents. *arXiv preprint arXiv:2110.08296*.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

Yi-Syuan Chen and Hong-Han Shuai. 2021. Meta-transfer learning for low-resource abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12692–12700.

Yulong Chen, Yang Liu, Ruochen Xu, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2023. Unisumm and summzoo: Unified model and diverse benchmark for few-shot summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12833–12855.

Cheng Deng, Tianhang Zhang, Zhongmou He, Yi Xu, Qiyuan Chen, Yuanyuan Shi, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, Zhouhan Lin, and Junxian He. 2023. K2: A foundation language model for geoscience knowledge understanding and utilization.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Pengcheng He, Baolin Peng, Liyang Lu, Song Wang, Jie Mei, Yang Liu, Ruochen Xu, Hany Hassan Awadalla, Yu Shi, Chenguang Zhu, et al. 2022. Z-code++: A pre-trained language model optimized for abstractive summarization. *arXiv preprint arXiv:2208.09770*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Daniel Kang and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation. *arXiv preprint arXiv:2004.14589*.

Harold D Lasswell. 1948. The structure and function of communication in society. *The communication of ideas*, 37(1):136–139.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*.

Inderjeet Mani and Mark T Maybury. 1999. *Advances in automatic text summarization*. MIT press.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *Mining text data*, pages 43–76.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Yiwen Shi, Ping Ren, Jing Wang, Biao Han, Taha ValizadehAslani, Felix Agbavor, Yi Zhang, Meng Hu, Liang Zhao, and Hualou Liang. 2023. Leveraging gpt-4 for food effect summarization to enhance product-specific guidance development via iterative prompting. *Journal of Biomedical Informatics*, 148:104533.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665, Toronto, Canada. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Summit: Iterative text summarization via chatgpt. *arXiv preprint arXiv:2305.14835*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023b. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. 2023c. MACSum: Controllable Summarization with Mixed Attributes. *Transactions of the Association for Computational Linguistics*, 11:787–803.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, Xin Chen, and Kai Yu. 2024. Chemdfm: Dialogue foundation model for chemistry.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2020. A joint sentence scoring and selection framework for neural extractive document summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:671–681.

A LLMs' Prompt to Generate Abstractive Data

The prompt for generating abstractive data needs to be customized according to the characteristics of the dataset to prevent the model from generating content that is too long or too short. Prompts are categorized into system prompts and user prompts. We employ a standardized system prompt: "*Generate a concise and coherent summary towards the given article and don't generate anything else. Make sure the summary is clear, informative, and well-structured.*". The corresponding data features and user prompts for different datasets are shown in Table 4 and 5.

Data Feature	avg. summary length of words/sentences: 51.08/2.71
User Prompt	Summarize the article in three sentences around 60 words.

Table 4: Data feature and user prompt for *CNN/DailyMail*

Data Feature	avg. summary length of words/sentences: 23.33/1.00
User Prompt	Summarize the article in one sentence around 35 words.

Table 5: Data feature and user prompt for *BBC XSum*

B Main Experiment Packages

Table 6 shows the pre-trained weights used in this paper.

Model	URL
BART	https://huggingface.co/facebook/bart-large
T5	https://huggingface.co/google-t5/t5-large
PEGASUS	https://huggingface.co/google/pegasus-cnn_dailymail https://huggingface.co/google/pegasus-xsum
LLaMA2	https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
Rouge-1/2/L	https://huggingface.co/docs/evaluate/index
BERTScore	https://github.com/Tiiiger/bert_score

Table 6: Models and pre-trained weight used in the paper