Compression Ratio Controlled Text Summarization

Stanford CS224N Custom Project

Zheng Wang
Department of Computer Science
Stanford University
peterwz@stanford.edu

1 Key information to include

- (Optional) External collaborators: None
- (Custom project only) Mentor: Yijia Shao
- (Optional) Sharing project: No

2 Abstract

We propose using large language models (LLMs) to build a summarization system that could adapt to user feedback. We propose to use compression ratio (the ratio between the summary length and the original article length) as a controllable toggle of summarization instead of a hard word limit. We found that although prompting works in generating high-quality summaries, it falls short in finely controlling the compression ratio of the summary because LLMs are prone to generate summaries of relatively fixed lengths. We have implemented a supervised fine-tuning pipeline with LoRA to fit the summary by their lengths. By fine-tuning Mistral-7B-Instruct using the pipeline, we achieve a better compression ratio control of summarizing complicated articles than direct prompting.

3 Introduction

Large language models (LLMs) like GPT-4 have revolutionized natural language processing (NLP) research by offering capabilities that exceed traditional summarization methods, even being preferred over human-generated summaries in some cases (Pu et al., 2023). Despite these advancements, creating summaries that meet specific user needs remains a challenge. LLMs can aid in this process but aren't always fully equipped to meet all user demands. This project explores the potential of LLMs in expository writing, which involves summarizing and organizing documents to produce new, valuable content. This is particularly useful for scholars who integrate information from various sources into a cohesive analysis, often leading to new discoveries.

The concept of an LLM-based expository writing assistant, as suggested in Shen et al. (2023), could follow a three-stage pipeline: identifying and extracting key evidence, synthesizing information, and facilitating text composition. While LLMs can augment the reading and extraction process and help in synthesizing and composing texts, they can also be prone to hallucination. Building systems that leverage LLMs for these purposes could significantly benefit fields such as academic research, medical studies, consulting, and education by accelerating the integration of new information with existing knowledge to produce novel insights.

In the realm of expository writing, we focus on customizing LLM-generated summaries to meet diverse user needs. For example, audience will have different preferences over the summary's brevity, with some users preferring concise summaries for quick assimilation and others requiring detailed summaries for in-depth understanding. Toggles over the summary's length could be a key building block of an effective expository writing system. Other summary customization examples include tailoring its content to user-specified topics, offering a more relevant and useful output.

In this project, we prioritize controlling the summary's length through the concept of *compression ratio*, which offers a more nuanced approach than setting a specific word limit or word count range. The compression ratio can be defined as the summary's length divided by the length of the data source. Using compression ratio as the controllable metric allows users to distinguish the varying significance of content within the source article and adjust the summary brevity with high flexibiltiy. This would address the needs of various applications such as academic writing and journalism.

Our goal is to develop a custom summarization model of knowledge intensive texts, such as Wikipedia articles. We reveal that straightforward ways to control summary brevity, such as prompting, may fall short of expectations. For zero-shot prompting, raw LLMs tend to have its own preference on generating summaries of specific lengths, regardless of how you integrate compression ratio information in the prompt. We have iteratively prompted GPT-4 to generate summaries of specific compression ratios, and GPT-4 have failed to perform that task on multiple occations. Multi-shot prompting will not work either, as the cost of feeding summarization examples to LLMs in every inference request is prohibitive.

We then resort to supervised fine-tuning (SFT) to solve this problem. We use GPT-4 to generate a fine-tuning dataset containing high quality summaries of different lengths of Wikipedia articles. We then fine-tune on Mistral-7B-Instruct (Jiang et al., 2023) via LoRA (Hu et al., 2021) for this dataset. Our best fine-tuned model deviates from the target compression ratio by only 0.0576 on average, performing much better than the zero-shot alternative. This shows that the SFT model is able to adapt to the compression ratio indicator in the prompt in most cases.

In short, here are our major contributions through this project: (1) We show that in-context learning, or prompt engineering, is insufficient at resolving custom summarization, such as brevity controls. (2) We find that LLMs tend to generate summaries with particular length preferences, which may be different from that of the users. (3) We use SFT via LoRA to create a model that performs better than prompting in brevity-controlled text summarization.

4 Related Work

Expository writing Systems. Expository writing has long been an important writing task (Kramer, 2021). Genuine efforts to explore automating expository writing started with the advent of large language models. Shen et al. (2023) raises the possibility that LLMs can enable an end-to-end expository writing application. Although currently LLMs can already generate good summarizations on-the-fly, building an expository writing system with user's customized needs is a very complex task that demands both very high reliability and very high degree of customization. For example, automating linking evidences would require a highly accurate semantic similarity search pipeline component. Progressing through expository writing systems would require work like this project to tune different LLM-powered modules and form building blocks of an expository writing system.

Brevity controlled summarization. In this project, we focus on controlling the summarization brevity of long articles such as Wikipedia text. Existing LLM summarization length restrictions focus on a hard word limit and generating particular texts. For example, Tang et al. (2023) uses prompts to generate hard-word limits on conversation (dialogue) summaries. In this project, we instead explore generating paragraphed summaries of longer and more complicated texts, and we found that prompt engineering is insufficient to achieve satisfactory results in this case.

Other attempts, such as Jie et al. (2023) use an RL reward model to penalize generating texts with lengths not falling in the desired range. However, Jie et al. (2023) still focuses on the hard-word limits. Hard word limits may be meaningless immediately after tokenization though, and they may not be flexible enough to capture the summarization task's difficulty when the source articles are of different lengths. This is why we control summarization brevity via compression ratios. Also, in this project we perform supervised fine-tuning instead of the more complex PPO-based RL methods.

5 Approach

Prompt Engineering. Our first attempt of controlling the summary's compression ratio is via direct prompting. The approach is similar to Tang et al. (2023), except that we are summarizing Wikipedia articles, which is much more challenging than summarizing a short conversation. We send to GPT-4

(OpenAI, 2024) a prompt that contains the Wikipedia article to summarize (which has length L), as well as our target compression ratio r. When the first query returns a summary of length l that does not satisfy $(r-\epsilon)L \leq l \leq (r+\epsilon)L$, where ϵ is the tolerable compression ratio error, we iteratively query GPT-4 indicating that its previous summarization response is too long or too short, steering GPT-4 towards our target compression ratio. This process is similar to how users will interact with a brevity-controlled summarization system in the real world.

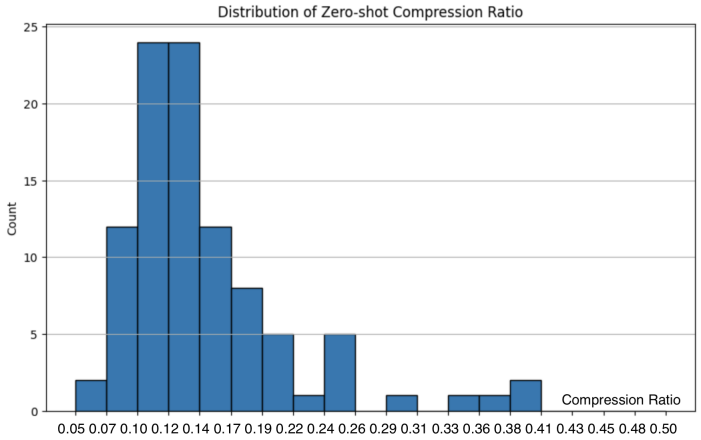

Figure 1: Zero-shot GPT-4 summary compression ratio distribution

We found that for Wikipedia articles of similar length, (1) the distribution of the summary compression ratio of GPT-4 is concentrated in a particular range (0.05 to 0.25), as shown in Figure 1. (2) It is difficult for GPT-4 to generate a summary with arbitrary compression ratio, as shown in figure 2. In this example, to acquire a target compression ratio of 0.2, we queried GPT-4 10 times and still did not get a summary with the desired compression ratio.

We further explored whether there is a correlation between the length of the original article and the difficulty for GPT-4 to generate a summary with a particular compression ratio. As shown in Figure 3, for shorter articles, it is easier to achieve a larger desired compression ratio (such as 0.200 when the word length is 900), and for longer articles, it is easier to achieve a shorter desired compression ratio (such as 0.075 when the word length is 1300). This shows that GPT-4 probably has a preference to generate a summary with a relatively stable length, instead of generating different summary lengths according to compression ratios. GPT-4 also, to a large degree, disregards the compression ratio hint we indicated in the prompt. This means that although GPT-4 may serve as a useful first shot summary generator, it is poor at generating brevity controlled summaries, in particular, compression-ratio controlled summaries.

Our method. Since naive prompting did not work as expected, we need to solve this problem with other methods. One of them is few-shot prompting. However, summarization of Wikipedia articles is a task that requires a huge amount of tokens to demonstrate. This would mean that inference via few shots is costly and even prohibitive for models with smaller context window. We propose to use supervised fine-tuning (SFT) to generate compression-ratio controlled summaries. To collect training data, we use GPT-4 to generate summaries of different lengths for a diverse set of Wikipedia articles. Because we are unable to fine-

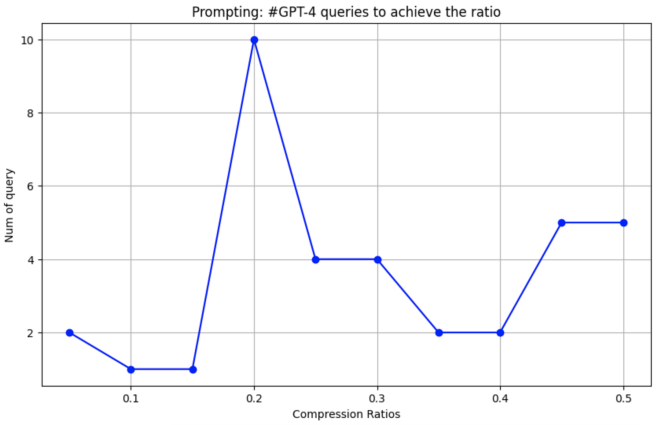

Figure 2: Number of GPT-4 queries to reach the target compression ratio for one example. Note that for compression ratio 0.2, we have tried 10 times and still could not get a summary with desired length. In this case $\epsilon = 0.05$.

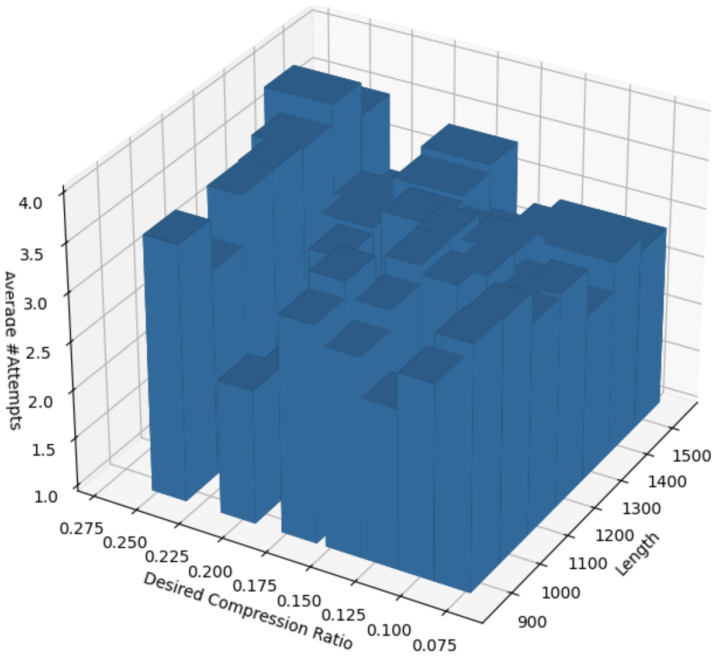

Figure 3: Relationship between source article length, desired compression ratio, and the average number of attempts for GPT-4 to finish summary generation. We will give up if GPT-4 fails to generate a summery with our desired compression ratio in 3 times. Smaller number of generation attempts means it is easier to achieve that compression ratio target.

tune on GPT-4 itself, we fine-tune a Mistral-7B with LoRA, with a prompt that takes the additional compression ratio as part of the instructions when generating summaries. By comparing the zero-shot performance of Mistral-7B and Mistral-7B instruct versus the LoRA fine-tuned version of these models, we can see to what extent the instruction fine-tuning would work at improving the performance of compression ratio controlled summarization.

Base on the DSPy library (Khattab et al., 2023) and the Llama-2 full fine-tuning codebase provided by the project mentor, we implemented the dataset creation and LoRA fine-tuning pipeline with Hugging Face PEFT library and Deepspeed (Rasley et al., 2020).

6 Experiments

Data. We collected 167 Wikipedia articles to summary pairs, split that into a training dataset containing 140 articles, and an evaluation dataset containing 27. The length of the original Wikipedia articles is between 800 and 1,500 words, filtering out articles that are too long or too short. This will also guarantee that the source, the prompt, and the generated summary will altogether fit in the context window of Mistral-7B (8192). We call this the **standard** dataset. In addition, we also collected 100 articles that are between 800 and 4,000 words. This dataset is closer to the target applications to summarize, but the source, the prompt, and the generated summary may fall out of the context window of Mistral-7B. We call this the **extended** dataset. Because the extended dataset may span out of LLM's context window, the extended dataset is of lower quality than the standard dataset.

To enable the LoRA model to learn the proper relationship between the compression ratio and the summary length, the distribution of the summary compression ratio should be relatively uniform as well. We created this dataset by iteratively prompting GPT-4 with different compression ratio targets. With this dataset, our model should generate a high-quality summary with a compression ratio close to our given target after supervised fine-tuning.

Evaluation Method. We use the ROUGE score (Lin, 2004) to evaluate the consistency of the summary. To measure the summary's brevity control, we will also evaluate the compression ra-

tio derivation as $\frac{1}{N}\sum_{i=1}^{N}\frac{|l_{actual}-l_{gold}|}{L}$, where l_{gold} is the length of a summary with the desired compression ratio, and L is the length of the source article.

Experiment Detail. We use LoRA with Rank 4 on top of Mistral-7B, only fine-tuning the query and the key layers. We fine-tune the model on one 80GB A-100 GPU. We only fine-tuned for 3 epochs. In the Results section below, we would show that fine-tuning for 10 epochs lead to overfitting. The fine-tuning process took about 1 to 2 hours. The learning rate follows the schedule defined in the left panel of 4.

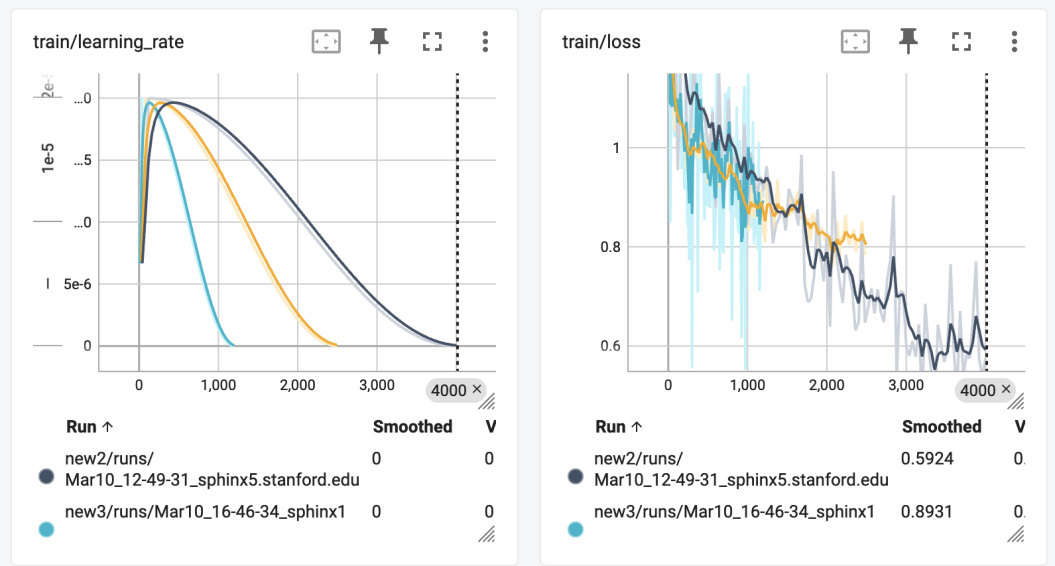

Figure 4: [Left] Our learning rate schedule. [Right] Our loss curve. Black line is the result of fine-tuning Mistral-7B on the standard dataset for 10 epochs. Yellow line is the result of fine-tuning Mistral-7B on the standard dataset for 3 epochs. Blue line is fine-tuning Mistral-7B on the extended dataset for 3 epochs.

Quantitative Results. As we can see from Table 1, on the Mistral-7B base model, fine-tuning led to a better result than prompting alone, in all the metrics we measure, including the Compression Ratio deviation metric. The same holds for Mistral-7B-Instruct as well. This means fine-tuning not only made the model generate summaries with better quality, but also able to make the model recognize the compression ratio hint in the prompt better. The better performance of Mistracl-7B-Instruct over Mistral-7B suggests that fine-tuning on general instruction tuned models may be better than the raw model at following user instructions.

As we can see from Table 2, on the extended dataset, fine-tuning still led to a better result than prompting alone, but excessive fine-tuning (for 10 epochs instead of 3) caused performance degradation, both in terms of fine-tuning quality and compression ratio deviation. This probably means that fine-tuning via LoRA for many epochs on a small dataset may lead to overfitting, because LoRA is only able to modify a very small amount of parameters in the neural network. Moreover, the SFT-3 model performance on the extended dataset is worse than on the standard dataset. This may be because the standard dataset only contains source articles of word length less than 1,500. This shows that a higher quality SFT dataset with less variation will lead to a more predictable in-distribution performance during evaluation.

Qualitative Analysis. Figure 5 shows the sample summary generations of the supervised fine-tuned Mistral-7B Instruct versus the summary generations of the original Mistral-7B Instruct. We can see that before fine-tuning, the Mistral-7B Instruct model generated the same summary to all the text, ignoring our compression ratio in the prompt. Although GPT-4 Turbo is different from Mistral-7B Instruct, this further supported our hypothesis that before fine-tuning on the compression ratio, large

Method	Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-lsum	CR Deviation
Prompting	Base	31.2101	11.0444	17.1503	22.7235	0.308
SFT-3	Base	51.0506	22.4898	32.8694	35.0922	0.0929
Prompting	Instruct	39.7568	15.0104	24.1671	26.0743	0.145
SFT-3	Instruct	**54.7672**	**23.4372**	**33.9436**	**37.8281**	**0.0576**

Table 1: Results on the **standard** dataset. Loss is the final value during training. SFT-3 means supervised fine-tuning for 3 epochs. CR means compression ratio. Base means fine-tuning on top of Mistral-7B, and Instruct means fine-tuning on top of Mistral-7B-Instruct. All generations are cut off at 800 tokens.

Method	Final Loss	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-lsum	CR Deviation
Prompting	-	22.2456	6.3367	11.8216	16.6108	0.140
SFT-3	0.9712	**45.8087**	**17.1481**	**25.6757**	**33.7372**	**0.079**
SFT-10	0.8038	44.1508	16.7056	23.5559	32.2294	0.0959

Table 2: Results on the **extended** dataset. Loss is the final value during training. SFT-3 means supervised fine-tuning for 3 epochs. CR means compression ratio.

language models have a preference on their desired summary length, so it is difficult to generate brevity controlled summaries via direct prompting.

After supervised fine-tuning though, the model is able to generate text with varied length according to the compression ratio, suggesting that the fine-tuning has been effective to relate compression ratio instructions at model's summary generation.

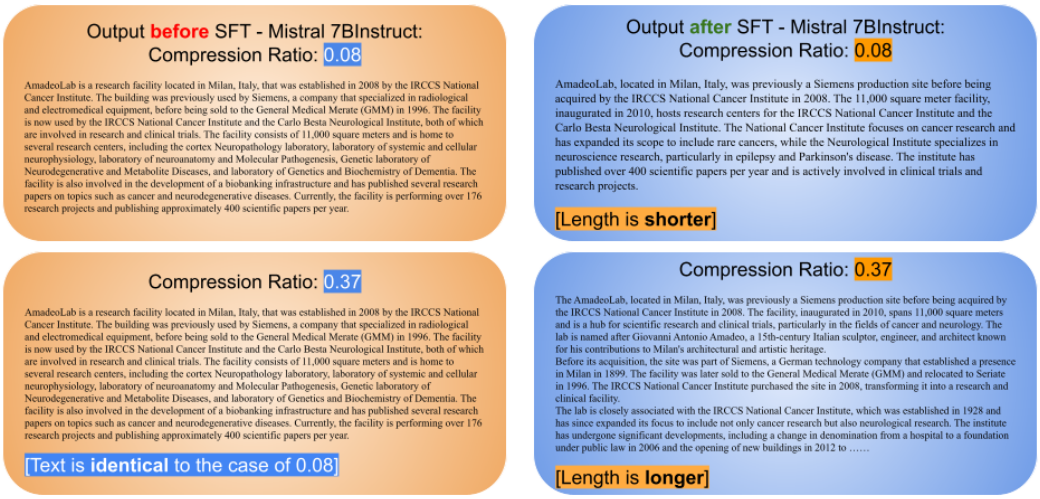

Figure 5: Generation example of SFT-3 Mistral-7B Instruct vs Prompting Mistral-7B Instruct.

7 Conclusion

A model successful at generating brevity controlled summaries is rewarding. Here, we list several takeaways from this project:

(1) The quality of the fine-tuning dataset matters. As shown by the difference between the **standard** and the **extended** dataset, a high quality standardized dataset could allow the model to generate well in distribution.

(2) Supervised fine-tuning, especially with LoRA, does not need too many epochs. Too many epochs may lead to overfitting, which is especially true for LoRA.

(3) For instruction specific tasks, fine-tuning on a instruction tuned model would have a better performance than fine-tuning on the base language model. This is because our prompt is ultimately instructions, and an instruction tuned model can follow instructions better.

Our project still has some limitations. Our best model still has a compression ratio deviation close to 0.06, suggesting that the brevity control of summaries is not fine grained. For the high quality standard dataset, the 1,500 word cap may be too low: a larger dataset with more diverse examples may lead to better SFT results.

As future steps of this project, we could further customize document summaries according to user's focus or intention. This would allow the model to generate tailored detailed summaries when the compression ratio is set low. These modules would help us to go one step further towards a better expository writing system.

References

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Renlong Jie, Xiaojun Meng, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Prompt-based length controlled generation with reinforcement learning.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Lindsay Kramer. 2021. Expository writing: Everything you need to know.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Zejiang Shen, Tal August, Pao Siangliulue, Kyle Lo, Jonathan Bragg, Jeff Hammerbacher, Doug Downey, and Joseph Chee Chang. 2023. Beyond summarization: Designing ai support for real-world expository writing tasks. *ArXiv*, abs/2304.02623.

Yuting Tang, Ratish Puduppully, Zhengyuan Liu, and Nancy Chen. 2023. In-context learning of large language models for controlled dialogue summarization: A holistic benchmark and empirical analysis. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 56–67, Singapore. Association for Computational Linguistics.

Appendix

7.1 Our GPT-4 prompt

We use this prompt to (1) perform the GPT-4 prompt engineering analysis in the "Approach" section. (2) generate the dataset for Mistral-7B Instruct supervised fine-tuning.

Initial prompt. You have access to a long Wikipedia page. Please write a summary for the page. The summary needs to be concise and contain all the important details in the Wikipedia page. Your summary needs to reach our target compression ratio. The compression ratio is the ratio between the length of the summary and the length of the original Wikipedia page. DO NOT REPEAT OUR INSTRUCTIONS IN THE SUMMARY! The original long Wikipedia page: [page] Our target compression ratio: [compression_ratio] Write a concise and detailed summary of the page:

Iterative prompt when GPT-4 generated a longer summary than expected. You have access to a Wikipedia page and a summary of the page. Write a shorter summary than the existing summary. The summary needs to be shorter and concise than the existing summary, but all the important information in the original Wikipedia page should stay in your more concise summary. Your summary needs to reach our target compression ratio. The compression ratio is the ratio between the length of the summary and the length of the original Wikipedia page. The existing summary has a compression ratio that is larger than what we expected. You can do better. DO NOT REPEAT OUR INSTRUCTIONS IN THE SUMMARY! The original Wikipedia page: [page] Our target compression ratio: [compression_ratio] The existing summary: [existing_summary] Your shorter summary:

Iterative prompt when GPT-4 generated a shorter summary than expected. You have access to a Wikipedia page and a summary of the page. Write a longer summary than the existing summary. The summary needs to be longer and more detailed than the existing summary, but all the important information in the original Wikipedia page should stay in your more detailed summary. Your summary needs to reach our target compression ratio. The compression ratio is the ratio between the length of the summary and the length of the original Wikipedia page. The existing summary has a compression ratio that is smaller than we expected. You can do better. DO NOT REPEAT OUR INSTRUCTIONS IN THE SUMMARY! The original Wikipedia page: [page] Our target compression ratio: [compression_ratio] The existing summary: [existing_summary] Your longer summary:

7.2 Our Mistral-7B (Instruct) SFT Prompt.

Below is an article. Write a summary that has a length roughly equal to the compression ratio times the original article's length. Article: [original_article] Ratio: [compression_ratio] Summary: