Ctyun AI at BioLaySumm: Enhancing Lay Summaries of Biomedical Articles Through Large Language Models and Data Augmentation

Ruijing Zhao[1,†,*], Siyu Bao[1,†,*], Siqin Zhang[1], Jinghui Zhang[1], Weiyin Wang[1], Yunian Ru[1],
[1]China Telecom Cloud Technology Co., Ltd
{zhaorj1,baosy,zhangsq20,zhangjh33,wangwy23,ruyn}@chinatelecom.cn
† Corresponding author

Abstract

Lay summaries play a crucial role in making scientific research accessible to a wider audience. However, generating lay summaries from lengthy articles poses significant challenges. We consider two approaches to address this issue: Hard Truncation, which preserves the most informative initial portion of the article, and Text Chunking, which segments articles into smaller, manageable chunks. Our workflow encompasses data preprocessing, augmentation, prompt engineering, and fine-tuning large language models. We explore the influence of pretrained model selection, inference prompt design, and hyperparameter tuning on summarization performance. Our methods demonstrate effectiveness in generating high-quality, informative lay summaries, achieving the second-best performance in the BioLaySumm shared task at BioNLP 2024.

1 Introduction

Biomedical publications serve as a critical channel for disseminating cutting-edge research findings on a wide range of health-related topics. While biomedical publications are essential for advancing medical knowledge and public health awareness, the technical terminology and lack of backgroud information often render them inaccessible to non-expert audiences(Guo et al., 2021). The BioLaySumm shared task addresses this need by developing effective models to generate lay summaries of biomedical articles aimed at non-expert audiences(Goldsack et al., 2024).

The challenge in the BioLaySumm shared task is to distill complex biomedical content into lay summaries that are both comprehensible and engaging to non-expert audiences. Large language models (LLMs) have shown remarkable capabilities in generating coherent and contextually accurate texts(Naveed et al., 2023), which could reformulate complex technical information into simpler narratives(Turbitt et al., 2023). Thus, LLMs are ideal for the generation of lay summaries. LLMs have witnessed the great advancement, each showcasing unique capabilities and specialized applications(Zhao et al., 2023), such as Mistral(Jiang et al., 2023), Qwen(Bai et al., 2023) and Llama(Touvron et al., 2023).

File	Key	Min	Max	Mean	Median
eLife	lay summary	225	893	478	473
	article	444	54,539	16,555	15,866
PLOS	lay summary	17	674	268	270
	article	1,046	37,770	10,289	10,029

Table 1: Token length statistics for the eLife and PLOS datasets, obtained using the Mistral tokenizer.

To tackle the challenge of lengthy articles in the BioLaySumm shared task, we consider two approaches: Hard Truncation and Text Chunking. We preprocess the data using these methods, apply data augmentation and prompt engineering, and finetune large language models on the task-specific data. We explore the effect of pretrained models, inference prompts, and hyperparameters on the quality of the generated lay summaries. Our experiments show that our approach effectively extracts key information and produces informative, easy-to-understand summaries.

2 Related Work

2.1 Large Languange Model Generation

Recent advancements in generation models have been dominated by the emergence of LLMs such as Mistral(Jiang et al., 2023), Llama(Touvron et al., 2023) and GPT-4(OpenAI et al., 2024). In the domain of biomedical summarization, LLMs have been adapted to interpret and summarize complex scientific texts, providing a foundation for tasks like BioLaySumm (Brown et al., 2020). Moreover, text chunking, an essential natural language pro-

[*]These authors contributed equally to this work.

Proceedings of the 23rd Workshop on Biomedical Language Processing, pages 837–844
August 16, 2024. ©2024 Association for Computational Linguistics

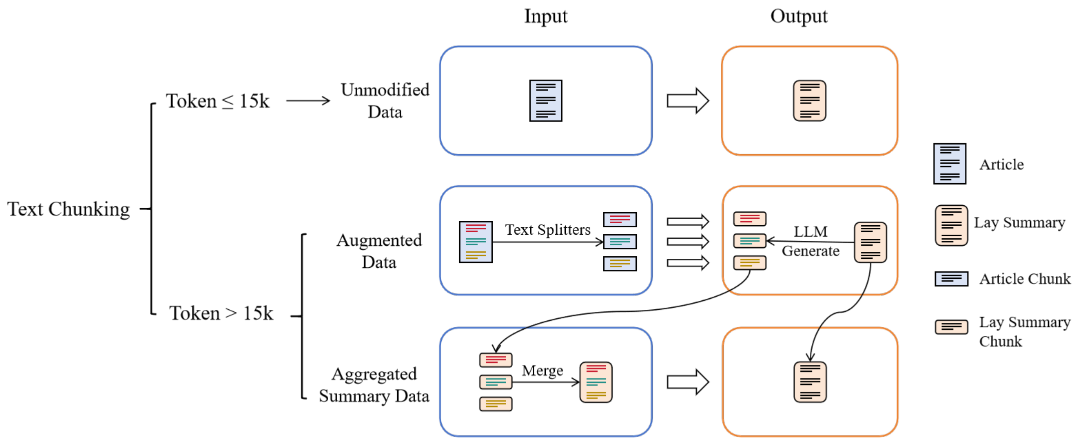

Figure 1: Text Chunking processes articles based on their token count. For articles with fewer than 15k tokens, the original content is preserved. Articles exceeding 15k tokens are divided into chunks, and the lay summary is generated using an LLM for each chunk. The generated lay summary chunks are then merged and used as input, with the original lay summary serving as the output.

cessing (NLP) technique, plays a critical role in BioLaySumm by breaking down large texts into manageable chunks(Reddy et al., 2023). This process enhances the accuracy of embedded content and improves important information retrieval, thereby enhancing the efficiency and quality of text retrieval and generation in the biomedical field.

2.2 Data Augmentation

Data augmentation (Shorten et al., 2021) in LLMs involves enriching the training dataset with artificially generated samples, which enhances the model's robustness and generalization capabilities. In biomedical summarization, data augmentation techniques such as back-translation (Sugiyama and Yoshinaga, 2019) and paraphrasing(Mi et al., 2022) have been used to expand the diversity of training examples, helping models to better handle a range of linguistic structures and terminologies found in medical texts (Li et al., 2022).

3 Data Preprocessing

3.1 Dataset

The dataset for BioLaySumm shared task is a combination of two biomedical datasets, PLOS and eLife(Goldsack et al., 2022). These datasets contain research articles and corresponding lay summaries written by experts .The diversity of these datasets presents a challenge for participants in developing models that effectively summarize biomedical literature for a general audience.

Between the two provided datasets, PLOS is larger, with 24,773 instances for training and 1,376 for validation, while eLife has 4,346 training instances and 241 validation instances.

3.2 Optimizing Input Article

Given the computational constraints, we limit the maximum context length to 15k tokens. Table 1 presents the token length statistics in the eLife and PLOS datasets. The statistics reveal that a considerable number of articles surpass the 15k token limit. We evaluate two approaches to address this challenge when applying Supervised Fine-Tuning (SFT) to adapt pretrained language models for specific tasks: Hard Truncation and Text Chunking.

Hard Truncation: This approach truncates articles, keeping only the first 15k tokens. It relies on the typical structure of articles, where crucial information is often presented initially. Truncating the latter part minimizes the loss of critical information while using only the provided data corpus. However, for longer articles, it may lead to information loss and potentially cause the model to generate content not present in the input.

Text Chunking: As shown in Figure 1, Text Chunking uses Langchain's Text Splitters* to divide articles into chunks of 15k tokens or less. This ensures the entire article is used in the SFT data. However, chunking introduces artificial boundaries within the text, which may disrupt the natural flow and context of the article, potentially impacting model performance. It also increases the number of training data entries, as a single entry may be

*https://python.langchain.com/v0.1/docs/modules/data_connection/document_transformers/recursive_text_splitter/

split into multiple chunks. This could result in longer articles having a disproportionate influence on the training process, as they contribute more chunks to the dataset.

We evaluate both methods on different datasets to determine the most optimal approach for each.

3.3 Data Augmentation

Hard Truncation does not introduce new content, but Text Chunking splits articles into fragments that do not match the original lay summaries. To address this issue, we use data augmentation with Mixtral 8x7B (Jiang et al., 2024) (hereafter Mixtral). Mixtral generates lay summaries for these fragments by finding the corresponding content from the full-text lay summary. It uses the original text as much as possible.

To include the full-text lay summary in the training data, we use the Mixtral-generated summaries as input and the original full-text summary as output. This incorporates the full-text summary into the training process for Text Chunking.

Data augmentation with Mixtral generates summaries that accurately correspond to the article fragments from Text Chunking. It also ensures the full-text summary is included in the training data.

3.4 Prompt Engineering for Data Segregation

For the Hard Truncation approach, a uniform prompt is used for all data entries. However, the Text Chunking method requires different prompts for three data types:

Unmodified Data: Articles not exceeding 15k tokens are retained directly and form the main portion of the training data. The prompt used for this data type is consistent with the one used during inference.

Augmented Data from Chunking: For articles split into chunks, the input text consists of the article chunk, while the output text is generated using Mixtral. A different prompt is employed during training to differentiate it from unmodified data.

Aggregated Summary Data: The outputs from augmented data from chunking are concatenated in the article's narrative order. This concatenated text serves as the input, and the original lay summary is used as the output. The prompt instructs the model to generate a concise lay summary from the overly long and redundant input.

The specific prompts used for each data type are presented in Table 6 of the Appendix.

4 Metrics

To thoroughly evaluate the quality of the generated lay summaries, we use a diverse set of metrics that capture various aspects of the summarization task:

Relevance: We use ROUGE (1, 2, and L) (Lin, 2004) and BERTScore (Zhang et al., 2019) to evaluate the relevance of the generated summaries to the original articles. Higher scores indicate better performance for these metrics.

Readability: To assess the readability of the generated summaries, we utilize several widely-used metrics: Flesch-Kincaid Grade Level (FKGL) (Kincaid et al., 1975), Dale-Chall Readability Score (DCRS) (Chall and Dale, 1995), Coleman-Liau Index (CLI) (Coleman and Liau, 1975), and LENS (Maddela et al., 2022). For FKGL, DCRS, and CLI, lower scores indicate better readability, while for LENS, higher scores are preferable.

Factuality: Ensuring the factual correctness of the generated summaries is crucial in the biomedical domain. We employ AlignScore (Zha et al., 2023) and SummaC (Laban et al., 2022) to measure the factual consistency between the generated summaries and the source articles. Higher scores on these metrics indicate better factual alignment.

5 Experiments

We conduct a series of experiments to investigate the impact of various factors on our lay summarization model's performance. Due to the PLOS validation set's size, we use the first 142 entries as our validation subset.

5.1 Impact of the Pretrained Model

We compare the performance of three pretrained language models: Qwen1.5-14B-Chat, Mistral-7B-Instruct-v0.2, and Meta-Llama-3-8B-Instruct. Each model is fine-tuned on the Hard Truncation dataset for one epoch with a learning rate of 1e-5 and a global batch size of 64. We use a complex prompt during inference, described in Section 5.2.

Table 2 shows the results. Meta-Llama-3-8B-Instruct achieves the highest LENS score but performs worse on other metrics. Qwen1.5-14B-Chat and Mistral-7B-Instruct-v0.2 exhibit comparable performance, with the latter having fewer parameters. Based on these findings, we select Mistral-7B-Instruct-v0.2 as our base model for subsequent experiments.

Model	ROUGE1	ROUGE2	ROUGEL	BERTScore	FKGL ↓	DCRS ↓	CLI ↓	LENS	AlignScore	SummaC
Qwen1.5-14B-Chat	0.4842	0.156	0.454	**0.8677**	**11.537**	9.559	**13.445**	54.865	0.7804	0.6876
Mistral-7B-Instruct-v0.2	**0.4959**	**0.1640**	**0.4654**	0.8672	12.054	**9.4289**	13.5558	52.0932	**0.7954**	**0.7070**
Meta-Llama-3-8B-Instruct	0.473	0.1464	0.4391	0.8581	12.0817	9.8036	13.5764	**66.8112**	0.739	0.6816

Table 2: Experiment results of different pretrained models. For FKGL, DCRS, and CLI, lower scores are better; for all other metrics, higher scores are better.

Prompt	ROUGE1	ROUGE2	ROUGEL	BERTScore	FKGL ↓	DCRS ↓	CLI ↓	LENS	AlignScore	SummaC
Simple Prompt	0.4804	0.1521	0.4514	0.8661	**11.936**	**9.3647**	**13.407**	**54.716**	0.7783	0.6716
Complex Prompt	**0.4959**	**0.1640**	**0.4654**	**0.8672**	12.054	9.4289	13.5558	52.0932	**0.7954**	**0.7070**
One-shot Prompt	0.4755	0.1496	0.4462	0.8652	12.104	9.4766	13.5491	54.232	0.7799	0.6694

Table 3: Experiment results of different inference prompts.

5.2 Impact of Inference Prompts

We investigate the impact of three distinct inference prompts on model performance: a simple prompt, a complex prompt, and a one-shot prompt. The specific prompts are detailed in Table 7.

Experiments using the Mistral-7B-Instruct-v0.2 model (Table 3) show that the complex prompt yields superior results compared to the simple prompt. The complex prompt improves relevance and factuality but slightly decreases readability. Surprisingly, the one-shot prompt underperforms the other prompts, possibly due to the lengthy example reducing content retention for the predicted sample. We use the complex prompt for subsequent experiments.

5.3 Impact of Hyperparameters

In the process of hyperparameter optimization, we drew inspiration from the experimental configurations employed in the Llama2 study. Our investigation focused on two critical hyperparameters: the number of training epochs and the learning rate. Specifically, we conducted a series of fine-tuning experiments using the Mistral-7B-Instruct-v0.2 model. The experimental design was as follows:

1. Single-epoch training with learning rates of 1e-5 and 2e-5.

2. Comparative analysis of single-epoch and dual-epoch training, both utilizing a learning rate of 1e-5.

This systematic approach allowed us to assess the individual and combined effects of epoch count and learning rate on model performance. By benchmarking against the Llama2 configurations, we aimed to leverage established best practices while adapting them to our specific task requirements. The results of these experiments provided valuable insights into the optimal hyperparameter settings for our fine-tuning process, enabling us to strike a balance between model performance and computational efficiency.

5.4 Impact of Data Augmentation

To address the challenge of articles exceeding 15k tokens, we developed and evaluated two distinct methods: Hard Truncation and Text Chunking. Hard Truncation preserves the original lay summary style but risks omitting content from the latter portions of the article. Conversely, Text Chunking ensures comprehensive inclusion of the entire article in the training set, albeit with the potential introduction of noise during data augmentation.

The application of these methods is contingent upon various factors. Hard Truncation may be more appropriate when less critical information is concentrated at the article's end or when sophisticated models for data transformation are unavailable. However, Text Chunking could potentially yield superior results when crucial content is distributed throughout the article.

To empirically assess the impact of these data processing methods, we fine-tuned separate models using datasets prepared with Hard Truncation and Text Chunking. The results, presented in Table 5, reveal that the Hard Truncation-trained model exhibits superior performance on the eLife dataset, while the Text Chunking-trained model demonstrates enhanced efficacy on the PLOS dataset. Leveraging these findings, we implemented an ensemble approach combining both models for our final submission. This strategy proved effective, securing 3rd place in relevance and 2nd place in the overall ranking of the competition.

6 Discussion

This paper introduces two methods for handling long input sequences in the BioLaySumm task and

Epoch	Learning Rate	ROUGE1	ROUGE2	ROUGEL	BERTScore	FKGL ↓	DCRS ↓	CLI ↓	LENS	AlignScore	SummaC
1	1e-5	**0.4959**	**0.1640**	**0.4654**	0.8672	**12.054**	**9.4289**	**13.5558**	52.0932	**0.7954**	**0.7070**
2	1e-5	0.4914	0.1549	0.4596	**0.8675**	12.217	9.576	13.58	**55.166**	0.76	0.6398
1	2e-5	0.4866	0.154	0.4544	0.866	12.551	9.7017	13.8178	52.575	0.7906	0.6587

Table 4: Experiment results of different hyperparameters.

Dataset	DataType	ROUGE1	ROUGE2	ROUGEL	BERTScore	FKGL ↓	DCRS ↓	CLI ↓	LENS	AlignScore	SummaC
eLife	Hard Truncation	**0.5153**	**0.1560**	**0.4904**	**0.8677**	9.9021	8.2115	11.6322	**62.9878**	0.6746	0.5714
	Text Chunking	0.4806	0.1451	0.4589	0.8642	**9.3846**	**7.9235**	**11.0592**	61.2874	**0.6961**	**0.5831**
PLOS	Hard Truncation	**0.4763**	0.1720	**0.4404**	0.8666	**14.2059**	**10.6464**	**15.4795**	**41.1988**	0.9162	0.8426
	Text Chunking	0.4748	**0.177**	0.4400	**0.8680**	14.644	10.77	15.864	40.742	**0.9558**	**0.8747**

Table 5: Experiment results of different data augmentation methods on eLife and PLOS dataset.

investigates the impact of various factors on generating lay summaries. Fine-tuning the Mistral-7B-Instruct-v0.2 model with specific settings yields strong performance.

Hard Truncation and Text Chunking's effectiveness varies depending on the target dataset. Hard Truncation may lose crucial information from later parts of long articles, potentially affecting summary completeness. Text Chunking, while preserving all content, introduces artificial boundaries that could disrupt context and lead to inconsistencies in generated summaries. Additionally, Text Chunking may result in longer articles having disproportionate influence on the training process. We use data augmentation with Mixtral, which generates summaries for text chunks. However, this approach may bias the model towards Mixtral's summarization style and introduce inconsistencies between fragment summaries and full-text summaries.

Future research could explore larger pretrained models and more sophisticated strategies for handling lengthy inputs. Section-specific summarization techniques could also improve performance.

Carefully designing inference prompts and selecting appropriate hyperparameters are crucial when fine-tuning pretrained language models for specific tasks. We hope our work inspires further research and contributes to developing effective tools for making scientific knowledge more accessible.

7 Limitation

In this study, we conducted a comprehensive analysis of various factors influencing model performance, including pre-trained models, hyperparameters, and data processing techniques. Our investigation, however, did not extend to examining the differential impact of distinct article sections on summary generation. This aspect warrants further exploration, as the introduction and conclusion sections often encapsulate the core content of an article and may hold greater significance for summarization, while body sections typically provide more granular details.

Additionally, to enhance the model's proficiency in specialized biological domains, future work could investigate the efficacy of incremental pretraining. This approach may potentially improve the model's ability to elucidate technical terminology in more accessible language, thereby enhancing the overall quality and comprehensibility of generated summaries.

These unexplored avenues present promising directions for future research, aimed at refining and advancing the performance of summarization models in specialized scientific domains, particularly in the field of biology.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jeanne Sternlicht Chall and Edgar Dale. 1995. Readability revisited: The new dale-chall readability formula.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Tomas Goldsack, Carolina Scarton, Matthew Shardlow, and Chenghua Lin. 2024. Overview of the biolaysumm 2024 shared task on the lay summarization of biomedical research articles. In *The 23rd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, Bangkok, Thailand. Association for Computational Linguistics.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. Making science simple: Corpora for the lay summarisation of scientific literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated lay language summarization of biomedical scientific reviews. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 160–168.

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. *Preprint*, arXiv:2310.06825.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *Ai Open*, 3:71–90.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2022. Lens: A learnable evaluation metric for text simplification. *arXiv preprint arXiv:2212.09739*.

Chenggang Mi, Lei Xie, and Yanning Zhang. 2022. Improving data augmentation for low resource speech-to-text translation with diverse paraphrasing. *Neural Networks*, 148:194–205.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Venkat praneeth Reddy, Pinnapu Reddy Harshavardhan Reddy, Karanam Sai Sumedh, and Raksha Sharma. 2023. IITR at BioLaySumm task 1:lay summarization of BioMedical articles using transformers. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 625–628, Toronto, Canada. Association for Computational Linguistics.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8(1):101.

Amane Sugiyama and Naoki Yoshinaga. 2019. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the fourth workshop on discourse in machine translation (DiscoMT 2019)*, pages 35–44.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Oisín Turbitt, Robert Bevan, and Mouhamad Aboshokor. 2023. MDC at BioLaySumm task 1: Evaluating GPT models for biomedical lay summarization. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 611–619, Toronto, Canada. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Prompts

In this sections, we delineate the specific content of the prompts employed in our experimental framework.

Data Type	Prompt
Unmodified Data	Generate a 300-400 word abstract for the given biology research article. Include research question, methods, main findings, implications, and conclusions. Use precise scientific terminology, logical structure, and active voice. Ensure clarity and accuracy.Here is the article:{input}. Please give me the clear abstract.
Augmented Data from Chunking	You will be given a section of a scientific article in the field of biology. Your task is to generate a concise and accurate summary of the key points and findings presented in this section. The summary should capture the main ideas, methods, results, and conclusions, while maintaining the scientific context and terminology used in the original text.Here is the article:{input}
Aggregated Summary Data	You will receive a summary of a biology research article generated by an AI model. However, the summary is too long and needs further refinement. Your task is to create a more concise version, focusing on the most critical information. The refined summary should:1. Maintain key findings, conclusions, and scientific context.2. Use precise, domain-specific terminology.3. Follow a logical structure highlighting main points.4. Aiming for 300-400 words.5. Omit unnecessary details while preserving the core message.6. Use clear, concise language for better readability.By adhering to these guidelines, create a highly refined summary that effectively conveys the essence of the original article.Here is the article:{input}

Table 6: Different prompts used for each data type in the experiments.

Prompt Type	Prompt
Simple Prompt	Please read the article given and write an easy-to-understand summary.Given article:{input}
Complex Prompt	Generate a 300-400 word abstract for the given biology research article. Include research question, methods, main findings, implications, and conclusions. Use precise scientific terminology, logical structure, and active voice. Ensure clarity and accuracy.Here is the article:{input}. Please give me the clear abstract.
One-Shot Prompt	Generate a 300-400 word abstract for the given biology research article. Include the research question, methods, main findings, implications, and conclusions. Use precise scientific terminology, a logical structure, and active voice. Ensure clarity and accuracy. The abstract should be written in the following format:{example}.Here is the full text of the research article to be summarized:{input}. Please provide a clear and professional abstract based on the article provided. Thank you!

Table 7: Prompt instructing the model to generate a concise lay summary from an overly long and redundant input summary