RAG-RLRC-LaySum at BioLaySumm: Integrating Retrieval-Augmented Generation and Readability Control for Layman Summarization of Biomedical Texts

Yuelyu Ji[1], Zhuochun Li[1], Rui Meng[2], Sonish Sivarajkumar[1], Yanshan Wang[1], Zeshui Yu[1], Hui Ji[1], Yushui Han[1], Hanyu Zeng [1], Daqing He[1]

[1]University of Pittsburgh, [2]Salesforce Research

Abstract

This paper introduces the RAG-RLRC-LaySum framework, designed to make complex biomedical research understandable to laymen through advanced Natural Language Processing (NLP) techniques. Our Retrieval Augmented Generation (RAG) solution, enhanced by a reranking method, utilizes multiple knowledge sources to ensure the precision and pertinence of lay summaries. Additionally, our Reinforcement Learning for Readability Control (RLRC) strategy improves readability, making scientific content comprehensible to non-specialists. Evaluations using the publicly accessible PLOS and eLife datasets show that our methods surpass Plain Gemini model, demonstrating a 20% increase in readability scores, a 15% improvement in ROUGE-2 relevance scores, and a 10% enhancement in factual accuracy. The RAG-RLRC-LaySum framework effectively democratizes scientific knowledge, enhancing public engagement with biomedical discoveries [1].

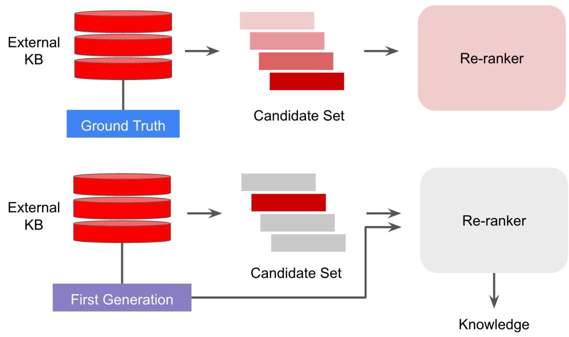

Figure 1: Knowledge Retrieval Augmented, with the trained re-ranker, can provide more relevant knowledge based on the first generation.

1 Introduction

Biomedical research encompasses crucial discoveries, ranging from everyday health concerns to significant disease outbreaks. Such studies are essential not only for scientists and doctors but also for journalists and the general public. However, the specialized and complex language typical in these studies often renders the content incomprehensible to those without a scientific background Thoppilan et al. (2022). To address this issue, the development of automated lay summaries have become increasingly important Goldsack et al. (2023b,a). This initiative aims to summarize the detailed aspects of biomedical research into summaries that are both comprehensible and devoid of complicated jargon. Although these systems show great potential, doubts about their accuracy are a major obstacle to their widespread useGabriel et al. (2020); Maynez et al. (2020); Yang et al. (2024); Li et al. (2023b, 2024); Wang et al. (2024). Our framework integrates specific external explanations for complex terms to further enhance content simplification. In response to the concerns about the integrity of summarized information, our framework employs a “knowledge retrieval” approach within the Retrieval-Augmented Generation (RAG) framework. This method uses a neural re-ranker to dynamically integrate trustworthy external knowledge sources like Wikipedia, ensuring that summaries are simplified, factually accurate, and contextually relevantLewis et al. (2020); Kang et al. (2024).

The architecture of the proposed RAG is illustrated in Figure 1. We also introduce a reward-based approach to overcome the limitations of traditional fine-tuning, which often produces summaries that have high ROUGE scores but are not actually readable to humans. This method fine-tunes the model by rewarding outputs that align with read-

[1]Our code and implementation details are available here: https://github.com/JoyDajunSpaceCraft/RAG-RLRC-LaySum

Proceedings of the 23rd Workshop on Biomedical Language Processing, pages 810–817
August 16, 2024.

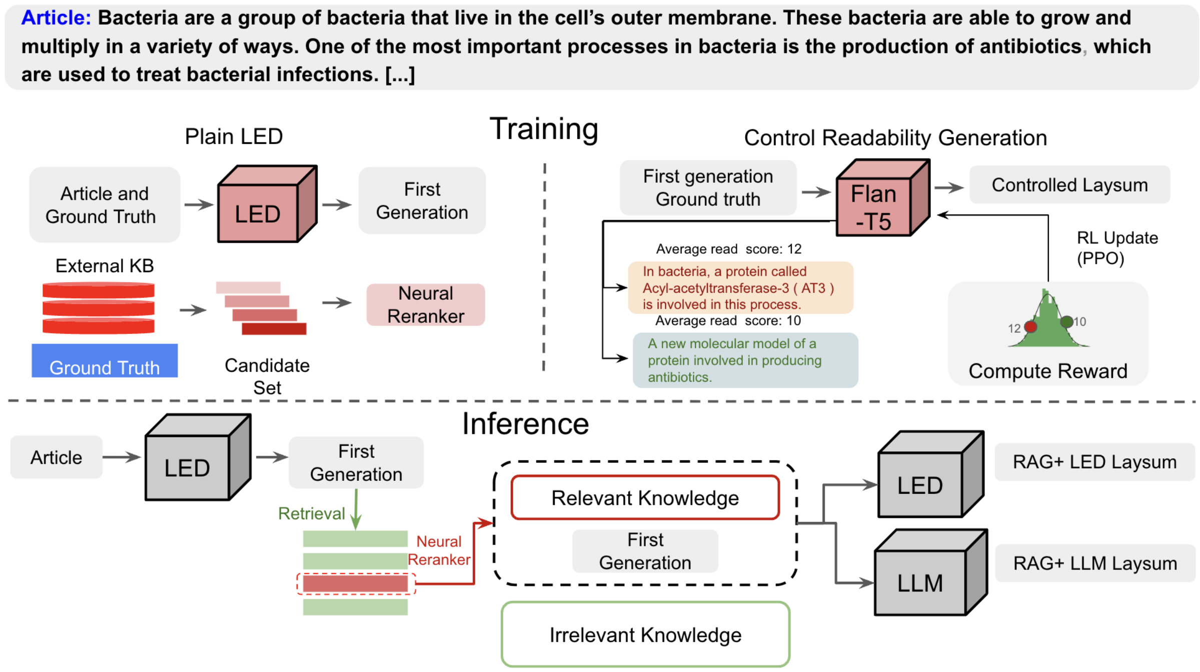

Figure 2: This figure illustrates the architecture of the proposed RAG-RLRC-LaySum model. During the training phase, we employ the Longformer Encoder-Decoder (LED) model as the backbone Beltagy et al. (2020). We enhance the model's capabilities through Wikipedia knowledge retrieval during inference. We utilize large language models (LLMs) such as ChatGPT and Gemini to further improve readability and enhance textual clarity by modifying prompts. For controlled text generation, readability scores are utilized to guide the model in generating expected outputs. The outputs of these scores are normalized to ensure text consistency and quality across generated texts.

ability metrics (Flesch-Kincaid Grade Level and Dale-Chall Readability Score Foster and Rhoney (2002); Ribeiro et al. (2023)). Unlike traditional supervised methods that might limit the model's adaptability, our approach encourages the model to alternative expressions to enhance clarity. The BioLaySumm challenge is a research competition that focused on developing and benchmarking models for generating lay summaries from complex biomedical literature[2]. Our method is ranked 11th on the leaderboard of the challenge Goldsack et al. (2024).

2 Methodology

First, we employ a Retrieval-Augmented Generation (RAG) solution that processes entire papers despite limited input capacity. Second, we improve summary quality by optimizing readability using relevant background information. The framework is illustrated in Figure 2.

2.1 Retrieval Augmented Generation (RAG)

Our RAG framework enhances keyword-based retrieval by using an initial lay summary generated by the model as a query. In the inference stage, it retrieves relevant descriptions from Wikipedia Ponzetto and Strube (2007) by the Pyserini Lin et al. (2021) index. However, retrieving relevant information from a large number of articles remains a challenge because the first generated summaries cannot work as effective queries. We initially use the ground truth as a query but switch to the first generated layman summary during inference for passage retrieval. However, there's a risk that the top-k passages may not be the most relevant for generating accurate summaries. Given a scientific document D with a set of candidate passages $K = \{k_1, k_2, ..., k_n\}$ from grounding sources, the RAG framework generates a lay summary S by maximizing the probability:

$$p(S|D, K) = \prod_{i=1}^{|S|} p(s_i|s_{<i}, D, K) \quad (1)$$

where s_i represents the i-th token in the summary, and $s_{<i}$ denotes the sequence of tokens preceding s_i. We use ColBERT Khattab and Zaharia (2020) and BGE-v2 Li et al. (2023a); Chen et al. (2024) as two different types of the neural re-ranker. The details about the trained re-ranker are in Ap-

[2]https://biolaysumm.org/

pendix B.

2.2 Reinforcement Learning for Readability Control (RLRC)

For details about the reranking model and sequence generation model training can be seen in Appendix A. The RLRC method inputs the first generation from the plain LED and uses the ground truth as the expected output. Our RLRC approach employs a reinforcement learning strategy to fine-tune the readability of summaries. We define a reward function $R(y, r^*)$ based on the desired readability level r^* that encourages the generation of text towards better readability, which is measured by the Flesch Reading Ease score r^*:

$$R(y, r^*) = 1 - \exp\left(-\frac{(R(y) - r^*)^2}{2\sigma^2}\right) \tag{2}$$

where $R(y)$ denotes the readability score of the generated summary y, and σ is a hyperparameter that controls the sensitivity of the reward function to deviations from the target readability score. Also, we leverage a Gaussian-based reward that strongly penalizes great variations in the readability Ribeiro et al. (2023).

We employ the Proximal Policy Optimization (PPO) Schulman et al. (2017) algorithm to optimize our RLRC model. The objective is to adjust the model's parameters θ by maximizing the objective function:

$$L(\theta) = \mathbb{E}_{(y,r^*)\sim p_{\theta_{\text{old}}}}\left[\left(\frac{p_\theta(y \mid D, r^*)}{p_{\theta_{\text{old}}}(y \mid D, r^*)}\right) R(y, r^*)\right] \tag{3}$$

Here, $p_{\theta_{old}}$ and p_θ denote the policy under the old and current parameters, respectively.

2.3 Large Language Models

We use the LLMs in two ways: first, as a paraphrasing tool during inference to refine initial generations, and second, for directly generating layman summaries. This implementation is built on Gemini-1.0-pro, developed by Google Team et al. (2023), which also serves as our baseline LLM. We aim to create readable summaries while incorporating as many input keywords as possible. We follow Gemini-1.0-pro's default settings, and the prompt details are described in Appendix C.

3 Experimental Settings and Results

3.1 Datasets and Evaluation

This study uses biomedical research articles from the PLOS and eLife datasets, which include both technical abstracts and expert-crafted lay summaries. The PLOS dataset contains 24,773 training and 1,376 validation instances, while the eLife dataset comprises 4,346 training and 241 validation instances Goldsack et al. (2022). We assess summarization quality using predefined metrics: Relevance is gauged by ROUGE Scores (ROUGE-1, ROUGE-2, ROUGE-L) Lin (2004) and BERTScore Zhang et al. (2019); Readability by the Flesch-Kincaid Grade Level (FKGL) Kincaid et al. (1975), Dale-Chall Readability Score (DCRS) Dale (1948), and Learnable Evaluation Metric for Text Simplification (LENS) Maddela et al. (2022); Factuality by Summac Laban et al. (2022) and AlignScore Zha et al. (2023).

3.2 Performance of Baseline Models

The Plain LED model, serving as our baseline, achieved ROUGE-L scores of 41.33 and 44.07. In contrast, the Plain retrieve+LED model, which integrates external knowledge through the BM25 retriever, slightly improved ROUGE-L scores to 47.02 and 47.21. This indicates that the incorporation of external knowledge slightly enhances the relevance of the summaries.

3.3 Effect of Neural Re-rankers

Further improvements were observed with the RAG+LED model, which incorporates a trained neural re-ranker, boosting the ROUGE-L scores to 49.68 and 49.79. This significant increase demonstrates that neural re-rankers are more precise in selecting relevant content, effectively enhancing the accuracy and relevance of the summaries.

3.4 Effect of Large Language Models

The RAG+ChatGPT and RAG+Gemini models, utilizing LLMs, achieved high FKGL readability scores of 9.93 and 9.25 respectively, but their ROUGE-L scores were lower at 39.59 and 39.20, indicating that LLMs can sometimes introduce irrelevant information. Similarly, the Plain Gemini model, which relies solely on an LLM, scored only 33.63 in ROUGE-L, demonstrating the challenges LLMs face in producing coherent and accurate summaries without mechanisms for precise content selection.

3.5 Effect of RLRC

The RAG+RLRC model, integrating reinforcement learning training strategies, achieved a ROUGE-L score of 47.24. It marked an improvement in

Table 1: Results on PLOS and eLife validation datasets. For the ↑ means, the higher, the better; for the ↓ means, the lower, the better. All best results are marked as bold. The RAG+different models represent the models that used neural re-ranker.

Method	Relevance				Readability				Factuality	
	Rouge1↑	Rouge2↑	RougeL↑	BERTScore↑	FKGL↓	DCRS↓	CLI↓	LENS↑	AlignScore↑	SummaC↑
PLOS										
Plain LED	45.96	**15.00**	41.33	**85.97**	15.17	12.26	16.42	54.96	**81.68**	**74.34**
Plain retrieve+LED	45.53	14.37	41.10	85.66	15.06	12.10	16.32	51.82	77.38	71.94
RAG+LED	45.64	15.37	**42.30**	85.21	15.22	11.93	15.92	53.42	76.57	72.83
RAG+ChatGPT	37.39	6.81	33.96	84.70	**11.21**	**10.37**	**12.50**	71.90	65.71	57.85
RAG+Gemini	38.89	8.74	35.11	85.12	11.33	10.48	13.38	**74.76**	68.40	58.88
Plain Gemini	44.67	13.36	40.26	85.87	15.71	11.84	17.98	62.64	74.18	52.82
RAG-RLRC	**46.58**	14.96	41.81	85.83	14.89	11.81	16.78	47.55	78.45	72.97
eLife										
Plain LED	47.02	12.52	44.07	**84.73**	10.52	9.33	11.49	73.45	**62.37**	60.12
Plain retrieve+LED	47.72	12.40	44.26	84.41	12.11	9.25	11.40	67.57	53.57	56.18
RAG+LED	47.69	12.41	44.34	84.41	11.99	9.25	**11.39**	67.95	53.89	55.65
RAG+ChatGPT	39.78	7.23	37.13	84.02	**9.58**	9.49	11.40	75.40	58.96	50.44
RAG+Gemini	39.90	9.04	36.97	84.29	**9.58**	9.65	12.47	**78.93**	62.91	55.81
Plain Gemini	22.60	3.22	20.85	80.81	16.38	12.72	24.18	52.44	53.19	44.97
RAG-RLRC	**47.91**	**12.65**	**44.96**	84.61	10.52	**9.11**	11.73	68.61	61.34	**60.40**
Average										
Plain LED	46.49	13.76	42.70	**85.35**	12.84	10.79	13.95	64.20	**72.02**	**67.23**
Plain retrieve+LED	46.62	13.38	42.68	85.03	13.58	10.67	13.86	59.69	65.47	64.06
RAG+LED	46.66	**13.89**	43.32	84.81	13.60	10.59	13.65	60.68	65.23	64.24
RAG+ChatGPT	38.59	7.02	35.55	84.36	**10.40**	**9.93**	**11.95**	73.65	62.34	54.15
RAG+Gemini	39.39	8.89	36.04	84.70	10.46	10.06	12.93	**76.85**	65.66	57.35
Plain Gemini	33.63	8.29	30.55	83.34	16.04	12.28	21.08	57.54	63.68	48.89
RAG-RLRC	**47.24**	13.80	**43.38**	85.22	12.70	10.46	14.25	58.08	69.89	66.68

factual accuracy, with a Summac score of 78.45 compared to the 73.44 of Plain LED. This highlights the effectiveness of reinforcement learning strategies in optimizing the text's factual alignment.

4 Related Work

Automatic summarization in the biomedical domain has been extensively studied Du et al. (2019); Krishna et al. (2020); Goldsack et al. (2023a); Devaraj et al. (2022). The primary challenge in this field is simplifying the content of original articles to make them comprehensible to laypersons. While Rosati (2023) supplement source documents to aid in generating more comprehensible summaries, and Devaraj et al. (2022) explore how text simplification impacts summary accuracy, introducing a taxonomy of error types and identifying omissions as a prevalent issue, these approaches often overlook the balance between simplification and factuality.

To enhance summary factuality, researchers incorporate factual knowledge from external sources during model training Mao et al. (2022), which has proven effective in improving accuracy. Rosati (2023) utilize Wikipedia to enrich summaries with additional knowledge, while Poornash et al. (2023) employ a trained re-ranker to select pertinent information, enhancing the factuality of summaries.

5 Conclusion and Future Work

The RAG-RLRC-LaySum framework effectively simplifies complex biomedical texts, enhancing readability and factual accuracy for lay audiences. It surpasses traditional models, offering new insights into the pivotal role of knowledge retrieval and readability optimization in scientific summarization. Future work will expand the framework's knowledge sources and refine how knowledge is utilized, potentially broadening its application across various scientific fields. This will further explore the integration of domain-specific knowledge to improve the precision and relevance of summaries.

6 Limitations

While the RAG-RLRC-LaySum framework shows promise, it has several limitations. The reliance on external sources like Wikipedia can introduce biases. The framework's computational complexity is high, making real-time applications challenging. Readability metrics like FKGL and DCRS may not fully capture readability for all audiences. Additionally, the generalizability to other domains beyond biomedical texts is uncertain. Lastly, evaluations based on automated metrics may not fully reflect user experience, highlighting the need for human evaluations. Future work should address these limitations by exploring diverse knowledge sources, optimizing efficiency, refining readability metrics, and conducting human evaluations.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

D Dale. 1948. The dale-chall formula for predicting readability. *Educational Research Bulletin*, 27:11–20.

Ashwin Devaraj, William Sheffield, Byron C Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2022, page 7331. NIH Public Access.

Nan Du, Kai Chen, Anjuli Kannan, Linh Tran, Yuhui Chen, and Izhak Shafran. 2019. Extracting symptoms and their status from clinical conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 915–925, Florence, Italy. Association for Computational Linguistics.

David R Foster and Denise H Rhoney. 2002. Readability of printed patient information for epileptic patients. *Annals of Pharmacotherapy*, 36(12):1856–1861.

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2020. Go figure: A meta evaluation of factuality in summarization. *arXiv preprint arXiv:2010.12834*.

Tomas Goldsack, Zheheng Luo, Qianqian Xie, Carolina Scarton, Matthew Shardlow, Sophia Ananiadou, and Chenghua Lin. 2023a. Overview of the biolaysumm 2023 shared task on lay summarization of biomedical research articles. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 468–477, Toronto, Canada. Association for Computational Linguistics.

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.

Tomas Goldsack, Zhihao Zhang, Chen Tang, Carolina Scarton, and Chenghua Lin. 2023b. Enhancing biomedical lay summarisation with external knowledge graphs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8016–8032, Singapore. Association for Computational Linguistics.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2024. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. *Advances in Neural Information Processing Systems*, 36.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Peter Kincaid, Robert P. Fishburne, 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.

Kundan Krishna, Sopan Khosla, Jeffrey P. Bigham, and Zachary C. Lipton. 2020. Generating SOAP notes from doctor-patient conversations. *CoRR*, abs/2005.01795.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023a. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Panfeng Li, Mohamed Abouelenien, and Rada Mihalcea. 2023b. Deception detection from linguistic and physiological data streams using bimodal convolutional neural networks. *arXiv preprint arXiv:2311.10944*.

Panfeng Li, Qikai Yang, Xieming Geng, Wenjing Zhou, Zhicheng Ding, and Yi Nian. 2024. Exploring diverse methods in visual question answering. *arXiv preprint arXiv:2404.13565*.

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.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*.

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

Qianren Mao, Jianxin Li, Hao Peng, Shizhu He, Lihong Wang, Philip S. Yu, and Zheng Wang. 2022. Fact-driven abstractive summarization by utilizing multi-granular multi-relational knowledge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1665–1678.

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

Simone Paolo Ponzetto and Michael Strube. 2007. An api for measuring the relatedness of words in wikipedia. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 49–52.

AS Poornash, Atharva Deshmukh, Archit Sharma, and Sriparna Saha. 2023. Aptsumm at biolaysumm task 1: Biomedical breakdown, improving readability by relevancy based selection. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 579–585.

Leonardo F. R. Ribeiro, Mohit Bansal, and Markus Dreyer. 2023. Generating summaries with controllable readability levels. In *Conference on Empirical Methods in Natural Language Processing*.

Domenic Rosati. 2023. Grasum at biolaysumm task 1: Background knowledge grounding for readable, relevant, and factual biomedical lay summaries. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 483–490.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Fali Wang, Runxue Bao, Suhang Wang, Wenchao Yu, Yanchi Liu, Wei Cheng, and Haifeng

Chen. 2024. Infuserki: Enhancing large language models with knowledge graphs via infuser-guided knowledge integration. *arXiv preprint arXiv:2402.11441*.

Qikai Yang, Panfeng Li, Zhicheng Ding, Wenjing Zhou, Yi Nian, and Xinhe Xu. 2024. A comparative study on enhancing prediction in social network advertisement through data augmentation. *arXiv preprint arXiv:2404.13812*.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

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*.

A Finetuning models

For training the Longformer Encoder-Decoder (LED) model Beltagy et al. (2020), we utilized the "allenai/led-base-16384" pre-trained checkpoint available on Huggingface's model hub. Our training setup included a configuration that processes 16,384 input tokens and generates outputs limited to 512 tokens. This training was conducted over the course of a single epoch.

In parallel, we employed BioLinkBERT-base Yasunaga et al. (2022) as the foundational language model for processing eLife and PLOS datasets, leveraging its specialized capabilities in understanding biomedical context.

Then, we designed a neural re-ranker based on the ColBERT Khattab and Zaharia (2020) and BGE-v2 Li et al. (2023a); Chen et al. (2024) scoring mechanism, which refines the results by evaluating the relevance of retrieved documents. The training for this re-ranker was tailored to accept inputs of up to 512 tokens, and it was fine-tuned to generate models by considering the top 5 most pertinent retrieval results. Futhermore, we define the Flan-T5-Large from huggingface, we use the model "google/flan-t5-large" as the base model. To make use of the control generation, we use the keywords in the article as the expected output to make sure the relevance.

A.1 Retrieval Augmented Generation

We conduct the experiment based on the model Longformer Encoder-Decoder (LED) Beltagy et al. (2020) which supports an input token length of 16,384 tokens. For the basic fine-tuning method, we find out in both the PLOS and eLife data that the re-ranker result will be a higher result in the Rouge-L and a lower score in the FKGL and DCRS score. In that case, indicate the lower the complexity of the description.

We use ColBERT Khattab and Zaharia (2020) and BGE-v2 Li et al. (2023a); Chen et al. (2024) as two different types of the neural re-ranker.

A.2 Reinforcement Learning for Readability Control (RLRC)

By utilizing various control levels for readability within the model-generated results, we focus on understanding how modifications to the readability scores, particularly the Flesch-Kincaid Grade Level (FKGL), impact the final summaries. The Flan-T5 model Chung et al. (2024) serves as the primary backbone for text generation. During the inference phase on testing data, where no ground truth is available for the reward mechanism, keywords are used as proxy indicators to ensure that the generated summaries accurately reflect the expected concepts.

In our model, we define two key mathematical expressions. The first is the Gaussian probability density function, used to estimate the likelihood of a given value within a normal distribution. The expression for this function is:

$$\text{calc\_nd}(value, mean) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(value - mean)^2}{2\sigma^2}\right) \tag{4}$$

This function is essential for assessing how far a data point deviates from the mean and is widely used in statistical analyses.

The second formula defines our reward function, which combines three different scoring metrics—readability score, BERTScore, and text length score—to comprehensively evaluate the quality of the text. The formula is as follows:

$$\text{reward} = w_r \cdot \text{normalized\_flesch\_scores} + w_b \cdot \text{all\_bertscore\_scores} + w_l \cdot \text{length\_scores} \tag{5}$$

Here, w_r, w_b and w_l are the weight factors for each scoring metric, adjusting the influence of each score in the overall assessment. By default, we set $w_r = 0.5, w_b = 0.3, w_l = 0.2$.

This weighted approach allows us to tailor the scoring criteria to different types of text analysis tasks, accommodating the multifaceted nature of text data.

B Retrieval Design

For the reranking of retrieved documents, we utilize the pyserini package Lin et al. (2021). Following the approach outlined by Rosati (2023), we employ $enwiki - paragraphs$ for background knowledge. We first retrieved 20 candidate paragraphs and then rerank the top 5 results.

B.1 Neural Re-ranker

In the provided Table 2, the performance trends across the eLife and PLOS datasets reveal that neural re-ranking methods (ColBERT and BGE) consistently outperform the traditional BM25 method. Notably, BGE shows a clear upward trend in accuracy from Top1 through Top20 in both datasets. Similarly, ColBERT's performance also exhibits an upward trajectory, although it remains below BGE, indicating a strong but second-tier efficacy among the tested methods.

Table 2: Accuracy for Neural Re-ranker.

Dataset	Method	Top1	Top5	Top20
eLife	BM25	10.32	42.13	65.24
	ColBERT	15.38	53.85	76.92
	BGE	18.53	60.19	78.52
PLOS	BM25	20.33	53.74	80.12
	ColBERT	26.09	57.97	84.06
	BGE	29.30	59.98	88.92

C Prompts

Table 3: One shot prompt for ChatGPT 4 and Gemini 1.0.

System: You are a layman rephrase; your goal is to rephrase the input and make it easier to read. For example: 'Diabetes is a condition in which the pancreas cannot produce enough insulin to feed the body. This is caused by a protein called proinsulin is an ingredient a group of molecules called cysteine thiols. The rephrased result should be: 'Diabetes is a condition where the pancreas doesn't produce enough insulin to meet the body's needs. This happens because of a protein called proinsulin, which consists of a group of molecules known as cysteine thiols.'
Input: Here is the original text I want you to help me to rephrase: {first generation}. Make it easier to read and retain as much of the biomedical phrase as possible and have a similar length as the original text.

Table 4: Prompt used for Gemini for article summarization.

I will give you a long article in biomedical publications, you should generate an abstractive summarization of this article in one single paragraph. I will also give you the keyphrases in this article, you should try to include as many keyphrases in your generated summarization as possible. The summarization is with an emphasis on catering to non-expert audiences through the generation of summaries that are more readable, containing more background information and less technical terminology. Keyphrases:{}, Article:{}.