Context Filtering with Reward Modeling in Question Answering

Sangryul Kim
KAIST AI
sangryul@kaist.ac.kr

James Thorne
KAIST AI
thorne@kaist.ac.kr

Abstract

Question Answering (QA) in NLP is the task of finding answers to a query within a relevant context retrieved by a retrieval system. Yet, the mix of relevant and irrelevant information in these contexts can hinder performance enhancements in QA tasks. To address this, we introduce a context filtering approach that removes non-essential details, summarizing crucial content through Reward Modeling. This method emphasizes keeping vital data while omitting the extraneous during summarization model training. We offer a framework for developing efficient QA models by discerning useful information from dataset pairs, bypassing the need for costly human evaluation. Furthermore, we show that our approach can significantly outperform the baseline, as evidenced by a 6.8-fold increase in the EM Per Token (EPT) metric, which we propose as a measure of token efficiency, indicating a notable token-efficiency boost for low-resource settings[1].

1 Introduction

The ability of language models to effectively understand and process long texts has become a critical requirement, particularly for question-answering (QA) applications (Beltagy et al., 2020; Feldman and El-Yaniv, 2019; Nan et al., 2021; Caciularu et al., 2022). However, several studies highlight the problem that even with a substantial amount of relevant context provided, the inclusion of irrelevant content within the context can adversely affect overall performance (Shi et al., 2023; Akimoto et al., 2023; Sauchuk et al., 2022; Oh and Thorne, 2023). The challenge often lies in distinguishing useful information from irrelevant details.

Our research tackles this issue by introducing a new approach to filter out unnecessary content, focusing on summarizing the key points through Reward Modeling. Specifically, we take note of the Direct Preference Optimization (DPO) method (Rafailov et al., 2023), which trains models using positive and negative feedback from datasets composed of pairs of "*chosen*" and "*rejected*" texts. We suggest employing this technique for filtering the context in QA tasks. We particularly focus on the process of inducing the chosen and negative datasets by paying attention to the presence or absence of specific information within the three essential elements required for the QA task: context, question, and answer. We investigate how the presence or lack of each piece of information impacts the reward modeling process, thus contributing to the development of an efficient context filtering model. Our method ultimately aims to enhance the efficiency of QA models by identifying and retaining only the most relevant information to the query, thereby improving performance.

We study context efficiency by introducing an

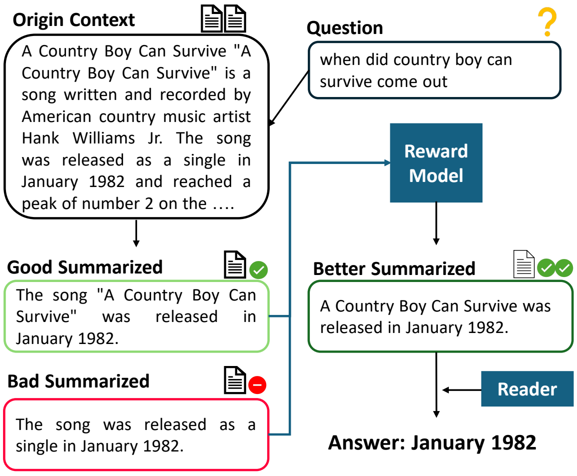
Origin Context
A Country Boy Can Survive "A Country Boy Can Survive" is a song written and recorded by American country music artist Hank Williams Jr. The song was released as a single in January 1982 and reached a peak of number 2 on the ....
Question
when did country boy can survive come out
Reward Model
Good Summarized
The song "A Country Boy Can Survive" was released in January 1982.
Bad Summarized
The song was released as a single in January 1982.
Better Summarized
A Country Boy Can Survive was released in January 1982.
Reader
Answer: January 1982

Figure 1: For an effective QA task, we conduct context filtering through the process of creating better summarization using a reward model. Simultaneously, we make it possible to discern which parts are helpful and which are filtered out by utilizing rewards extracted from the data.

[1] Code and datasets are available at https://github.com/xfactlab/coling2025-context-filtering

Proceedings of the 31st International Conference on Computational Linguistics, pages 11048–11055
January 19–24, 2025. ©2025 Association for Computational Linguistics

EM Per Token (EPT) metric and use it for comparisons between models. This allows us to evaluate the trade-off between context length and the answer Exact Match (EM) score. This is motivated by the fact that more context induces diminishing returns (Izacard and Grave, 2021) and comes with performance overheads.

2 Backgrounds

2.1 Knowledge Refinement

For open-domain question answering, the prevailing trend in research is focused on *retrieving* the correct context from a corpus of knowledge to condition a *reader* (Chen et al., 2017; Karpukhin et al., 2020; De Cao et al., 2021; Petroni et al., 2021). Simultaneously, language models have been used to augment question information (Chen et al., 2023) context generation (Yu et al., 2022), and summarize / paraphrase retrieved information (Xu et al., 2023; Lee et al., 2023). However, recent studies claim that the presence of irrelevant information within the context can lead to a decrease in the performance of the model in a process called *detrimental retrieval* (Sauchuk et al., 2022; Oh and Thorne, 2023; Shi et al., 2023; Akimoto et al., 2023). Therefore, there is a need for research on models that can filter detrimental content from the retrieved context while aggregating only the essential information.

2.2 Reward Modeling

Many methodologies for training LLMs have been developed to align models to user preferences: models are steered away from generating toxic or unhelpful responses. Many methods are derived from the Bradley-Terry model of competition (Bradley and Terry, 1952), using preference pairs containing chosen and rejected responses. In particular, Reinforcement Learning from Human Feedback (Ouyang et al., 2022, RLHF) incorporates human evaluations for training a reward model that is used to score responses, guiding fine-tuning a text generation model to align with user preferences. However, a drawback of this approach is the need to collect human preferences for training, which can be costly. Additionally, Proximal Policy Optimization (PPO), commonly used in RLHF, is sensitive to hyperparameter settings. Improper tuning can lead to training instability or divergence (Schulman et al., 2017; Hsu et al., 2020). Overcoming some of these limitations, Rafailov et al. (2023) proposed Direct Preference Optimization (DPO). In DPO, the model itself serves as the source of the reward function. Let y_w be the preferred output and y_l be the rejected output for a given given task prompt x and denote the dataset $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^{N}$. We can denote the loss $\mathcal{L}_R(r_\phi, D)$ with the reward function $r_\phi(y, x)$ as below (Rafailov et al., 2023):

$$-\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right] \tag{1}$$

In the QA task, we assume that items with a very high likelihood of containing the answer and whose surrounding context is closely related to the answer are "chosen", while those with a low likelihood of containing the answer and whose surrounding context is likely unrelated to the answer are "rejected". Therefore, we design experiments to find a method that filters only the context necessary for the QA model by conducting training in a way that increases the margin between "chosen" and "rejected" in association with the DPO reward loss.

3 Context Filtering

Our experiments are performed studying models for open-domain question-answering. We adopt a *summarize-then-read* pipeline structure adapting the model structure from Inoue et al. (2021). The pipeline consists of an abstractive summarization model and a question answering model. During the summarize phase, we compare SFT and DPO fine-tuning to determine how efficiently each model summarizes. In the question answering phase, we compare the ability to find the correct answers within the context that has been filtered out through summarization. We use the FLAN-T5-XL model (Chung et al., 2022)[2] from Hugging Face (Wolf et al., 2020) (3B parameters) throughout all experiments. This seq2seq encoder-decoder model is pre-trained for abstractive summarization and captures the main point of the context. All specific settings, such as prompts and model templates used in the experiments, can be found in Appendix B.

3.1 Data Generation

Our experiments are conducted on three question answering datasets: SQuAD v1.1 (Rajpurkar et al., 2016), Natural Question (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017). Retrieved contextual information is selected from DPR (Karpukhin et al., 2020)[3] for NQ and TQA.

[2] https://huggingface.co/google/flan-t5-xl

[3] https://github.com/facebookresearch/DPR

For NQ and TQA, instances where the answer to the question is not within the top-1 relevant context are excluded. We split the training set of SQuAD to create an additional validation dataset and use the existing validation set as the test set. The datasets postfixed with "r" in the table are those that have undergone this process. Detailed information is in Appendix C.

For the QA, the task aims to find the corresponding answer A for a given question Q and a context C. We can establish three strategies for summarization through the missing combinations in the necessary information denoted as the tuple $I = (Q, A, C)$. We can construct prompts with different combinations of information:

Type 1 consists of $I_1 = (Q, A, C)$, containing all the information (question, answer, and context), aiming to achieve the best possible summarization for comparison and for the highest likelihood of generating correct answers for pairwise training. However, this would not be realistic in an unseen test setting.

Type 2 consists of $I_2 = (Q, C)$, and the goal is to summarize the parts related to the question in the context without providing information about the answer. This may be most realistic for application to unseen questions and represents the task signature of the model we would aim to train.

Type 3 is composed of $I_3 = (A, C)$, focusing on summarizing or extracting related parts in a more lexical aspect, ensuring that the model includes the answer, regardless of the context. We use the outputs generated from the various types of prompt configurations in the following experiments.

3.2 SFT Summarizer Training

The objective of Supervised Fine Tuning(SFT) is to fine-tune the base model to generate the output summary O_1 created with the Type 1 prompt when given a context and question (Type 2) as input, learning the policy $\pi^{sft}(O_1|I_2)$. Through this process, the fine-tuned summarization model π^{sft} should be able to utilize all the information included in $I_1 = (Q, A, C)$ to produce the best summarization results given $I_2 = (Q, C)$.

3.3 DPO Summarizer Training

For the DPO Training, we require pairs of answers (y_1, y_2) and need to determine which summary would be y_w and y_l satisfying $y_w \succ y_l \mid x$ (Rafailov et al., 2023). Typically human labelers or an LLM determine y_w and y_l, but in this architecture scenario, we assume that outputs from the base model with two types prompts; (Type 2, Type 3) can be candidate of the y_l, denoted as O_2, O_3 respectively. This is because we assume that the outputs from Type 2 and Type 3, which were created with missing information, are less preferred than those from Type 1. We aim to understand how the lack of information in each of Types 2 and 3 affects the reward model through DPO training compared to Type 1. Hense, we construct two different DPO model $\pi^{dpo}_{O_1,O_2}$, $\pi^{dpo}_{O_1,O_3}$. We use Hugging Face TRL (von Werra et al., 2020) for DPO Training.

3.4 Reader Training

We study the impact on reducing context through summarization, and thus the reduced context length, on the downstream reader. We train the reader to generate answers using the filtered context, which is achieved by summarizing each dataset with the Type 1 trained summarization models. For baseline evaluation, using Type 1 will determine how well each model can deliver answers without loss of information in the "read" phase of the *summarize-then-read* pipeline.

3.5 Evaluation

To assess the reader's ability to generate answers using summarized contexts, we use the exact match (EM) score and unigram F1 metrics (Rajpurkar et al., 2016). For the NQ and TQA datasets, where there are multiple possible answers, we initially filtered based on whether the first correct answer was included in the context, treating the first answer as correct if it matched the given answer. Additionally, we calculate the number of tokens required to find the answer in the context using our proposed EM Per Token (EPT) metric. EPT serves as an indicator of the efficiency of the given context, c:

$$EPT(c, y^*, \hat{y}) = \frac{EM(y^*, \hat{y})}{|c|} \qquad (2)$$

Furthermore, to evaluate the performance of the summarization itself, we introduce a metric called Inclusion Rate of Answer (IRA) shown in Table 2. The IRA can verify whether the target answer is fully contained within the given context. If the answer is exactly present, it is evaluated as 1; if it is partially missing or not included, it is evaluated as 0. This metric assesses how well the summarization is done in terms of leaving room for the reader to

Dataset	**NQ$_r$**				**TQA$_r$**				**SQuAD$_r$**			
Model	**EM**	**F1**	**Tok Len**	**EPT**	**EM**	**F1**	**Tok Len**	**EPT**	**EM**	**F1**	**Tok Len**	**EPT**
Origin	59.59	67.71	147.30	0.40	77.38	83.44	152.41	0.51	68.32	82.97	179.83	0.38
SFT	55.21	63.28	29.99	**1.84**	69.68	75.49	28.88	2.41	59.66	76.07	30.26	1.97
DPO$_{O_1,O_2}$	49.87	58.68	41.92	1.19	66.06	72.80	39.52	1.67	48.79	61.85	35.61	1.37
DPO$_{O_1,O_3}$	52.83	59.83	29.97	1.76	62.29	68.16	18.20	**3.42**	55.40	69.28	21.46	**2.58**

Table 1: Performance comparison across models on different datasets. EM: Exact Match, F1: F1-score, Tok Len: Token Length, EPT: Efficiency per Token.

Dataset	**NQ$_r$**	**TQA$_r$**	**SQuAD$_r$**
Model		**IRA**	
Original	100	100	100
SFT	75.93	78.76	89.94
DPO$_{O_1,O_2}$	71.94	77.27	66.93
DPO$_{O_1,O_3}$	68.60	65.91	73.88

Table 2: The ratio indicating whether the target answer is included within the context summarized by each model when the original context is summarized.

find the answer, essentially checking the potential for finding the answer before the reader phase. This metric evaluates the summarization process itself, independent of the subsequent reading phase.

4 Impact of Context Filtering

Trade-off between Token Length and Accuracy Metrics. This experiment focuses on extractive QA datasets, where answering a question requires not only identifying the correct answer and its relevant context but also understanding the relationships between different parts of the context that contribute to the answer. By comparing the IRA scores in Table 2 with the EM scores in Table 1, we observe that context length does not directly correlate with accuracy. To further investigate, we analyze the contribution of individual tokens to the EM score using the previously defined EPT metric.

Our analysis, conducted with both the SFT and DPO models, demonstrates that models retain accuray with reduced context lengths. For the NQ dataset, reducing the token length to just 20% of the original retains 92% of the initial performance. Similarly, for the TQA dataset, a token length of 12% preserves 80% of the performance, while for the SQuAD dataset, 12% of the token length achieves 81% of the original performance. These results highlight the potential for substantial token reduction without a severe loss in accuracy and underscore potential efficiency gains.

From another perspective, we can observe that the filtering process in SFT and DPO also leads to the loss of information related to the answer span. This can be interpreted in two ways. Firstly, when examining the prompt *"Summarize below context into one sentence..."* used in SFT and DPO, the answer might have been deemed relatively less important in the process of reducing it to one sentence from a summarization standpoint. This suggests that some dependency on the prompt and the model remains, indicating room for improvement. Secondly, our approach to filtering the context did not simply involve lexically cutting off existing content or directly extracting it (Wang et al., 2023); rather, we restructured it through a prompt, transforming filtering into a summarization task.

Despite this, the results in Table 1 still reveal that the trade-off between token length and EM and F1 metrics is favorable; the efficiency gained from filtering outweighs the slight loss in accuracy, indicating benefits of filtering.

Comparison with Different Reward Model Strategies. DPO is known to enable credit assignment down to the token level (Rafailov et al., 2024). Therefore, after training, we can indirectly estimate how it influenced tokens by analyzing the metrics in Table 1 and the EM rates based on length. During the Data Generation phase (Section 3.1), we empirically observe that Type 2 outputs, O_2, are summarized in a form that provides short answers about the context and question, while the Type 3 outputs, O_3, are more reflective of the lexical elements surrounding the answer in the context. Furthermore, the presence of the answer in the prompt for Type 3 induces generations that are more similar to Type 1 compared to Type 2. Therefore, during DPO training, the DPO$_{O_1,O_2}$ model is designed to produce longer outputs that are more similar to those of Type 1, whereas responses from DPO$_{O_1,O_3}$ are shorter outputs centered around elements present in Type 1 but absent in Type 3 outputs, indicating the intended direction of the training results.

5 Conclusions

In this work, we aim to construct an efficient QA system by filtering out unnecessary information from the context. By introducing the EPT metric, we assess the efficiency of the context in QA tasks. The results demonstrate that using the DPO model with reward modeling and its underlying SFT model for filtering is more efficient (per token) than using the full original context. In our future experiments following this study, we plan to apply the reward modeling concept developed here to conduct experiments on effective retrieval models. Moving beyond filtering within a single context, we aim to explore filtering across multiple contexts and incorporate a rewarding model that uses reader performance as a reward. Our goal is to build an integrated Information Retrieval (IR) system that enhances the overall efficiency and effectiveness of the retrieval process.

Limitations

In this work, we focus on deriving an efficient context filtering model through reward modeling. The essence of the reward lies in the base model, such as the SFT model evaluating the reward, and the pairs of chosen and rejected for DPO training. We generate data relying on the model's parameters by providing complete information and incomplete information according to each type, without human evaluation, and use this data for our experiments. During this process, we identify the introduction of some unintended biases, leading to biased results and inconsistent outcomes across the datasets. Given the nature of the research, it is crucial to generate data that aligns with the intended purpose during the training data generation stage. Therefore, when conducting additional follow-up experiments with this idea, we believe it is necessary to establish proper metrics at the data level and a verification process for evaluating the efficiency and assessing context filtering in QA tasks, separate from the efficiency evaluation.

Moreover, for our experiments, we used datasets that were modified and reduced from existing sources like NQ, TQA, and SQuAD after undergoing specific processing to suit our experimental needs. We believe that adding more diverse datasets, including those covering multi-hop QA (Yang et al., 2018) and long context QA (Fan et al., 2019), would allow for deeper interpretations.

Ethics Statement

In our experiments, we utilize the FLAN-T5 (Chung et al., 2022) model, which is a T5 (Raffel et al., 2020) model further enhanced with instruction tuning. Therefore, during the training process, the pre-trained parameters could indirectly or directly influence the outcomes, leading to uncontrolled generations that might result in content that is ethically or socially problematic. Consequently, when disclosing such research to the public, it's crucial to be aware of these potential risks and consider implementing engineering and research-based mitigative measures to prevent such issues. We utilize ChatGPT in the process of writing papers, using it for grammar correction and refining sentences. During this process, expressions may be included that differ from the intended meaning.

Acknowledgments

This work was partly supported by Institute for Information & communications Technology Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Support Program(KAIST) and National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00406715, AI for All: A Social Platform for Barrier-free AI). This work was also supported by the Artificial intelligence convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

References

Kosuke Akimoto, Kunihiro Takeoka, and Masafumi Oyamada. 2023. Context quality matters in training fusion-in-decoder for extractive open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11711–11729, Singapore. Association for Computational Linguistics.

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

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. 2022. Long context question answering via supervised contrastive learning. In *Proceedings* *of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2872–2879, Seattle, United States. Association for Computational Linguistics.

Andong Chen, Yuan Sun, Xiaobing Zhao, Rosella Galindo Esparza, Kehai Chen, Yang Xiang, Tiejun Zhao, and Min Zhang. 2023. Improving low-resource question answering by augmenting question information. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10413–10420, Singapore. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy. Association for Computational Linguistics.

Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. 2020. Revisiting design choices in proximal policy optimization. *arXiv preprint arXiv:2009.10897*.

Naoya Inoue, Harsh Trivedi, Steven Sinha, Niranjan Balasubramanian, and Kentaro Inui. 2021. Summarize-then-answer: Generating concise explanations for multi-hop reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6064–6080, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Yejoon Lee, Philhoon Oh, and James Thorne. 2023. Knowledge corpus error in question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9183–9197, Singapore. Association for Computational Linguistics.

Feng Nan, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Kathleen McKeown, Ramesh Nallapati, Dejiao Zhang, Zhiguo Wang, Andrew O. Arnold, and Bing Xiang. 2021. Improving factual consistency of abstractive summarization via question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6881–6894, Online. Association for Computational Linguistics.

Philhoon Oh and James Thorne. 2023. Detrimental contexts in open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11589–11605, Singapore. Association for Computational Linguistics.

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.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From r to Q^*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Artsiom Sauchuk, James Thorne, Alon Halevy, Nicola Tonellotto, and Fabrizio Silvestri. 2022. On the role of relevance in natural language processing tasks. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1789.

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

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *Preprint*, arXiv:2310.04408.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

A Hyperparameters

In our experiments, we conduct three training phases: SFT training, DPO training, and reader training. For each experiment, we utilize one NVIDIA A100 80GB or NVIDIA H100 80GB GPU. We set the number of training epochs to 3, and both the training and evaluation batch sizes to 4, with a gradient accumulation step of 32. The optimizer used is "paged_adamw_32bit," with a learning rate of 2e-4, and we employ a "cosine" type learning rate scheduler.

B Prompts and Templates

B.1 Prompts for Data Geneartion and SFT Training

[Type 1]

```
Summarize below context into one
sentence according to fit the
following context, question and
answer.
Context: {context}
Question: {question}
Answer: {answer}
Output:
```

[Type 2]

```
Summarize below context into one
sentence according to fit the
following context and question.
Context: {context}
Question: {question}
Output:
```

[Type 3]

```
Summarize below context into one
sentence according to fit the
following context and answer.
Context: {context}
Answer: {answer}
Output:
```

B.2 Prompts for Reader Training

[Train Input]

```
Given the context and question,
predict the answer to the question.
Context: {context}
Question: {question}
Answer:
```

[Train Output]

```
{target answer}
```

C Details of Datasets

Dataset	NQ$_r$	TQA$_r$	SQuAD$_r$
Split	**Number of Datasets**		
Train	79,618 (43,032)	78,785 (15,759)	80,000
Validation	8,757 (4,164)	8,837 (1,534)	7,599
Test	3,610 (1,554)	11,313 (1,883)	10,570

Table 3: The specific numbers of data used in the entire pipeline. For NQ and TQA, the numbers in parentheses indicate the actual amount of data involved.

The number of datasets involved in the experiment is depicted in table 3.