Context Filtering with Reward Modeling in Question Answering

Anonymous ACL submission

Abstract

Question Answering (QA) in NLP is a task that involves finding answers using information within a context. 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 enhance performance beyond the baseline, as evidenced by the EM Per Token (EPT) metric, suggesting a notable performance boost from the perspective of token efficiency for the low resource settings.

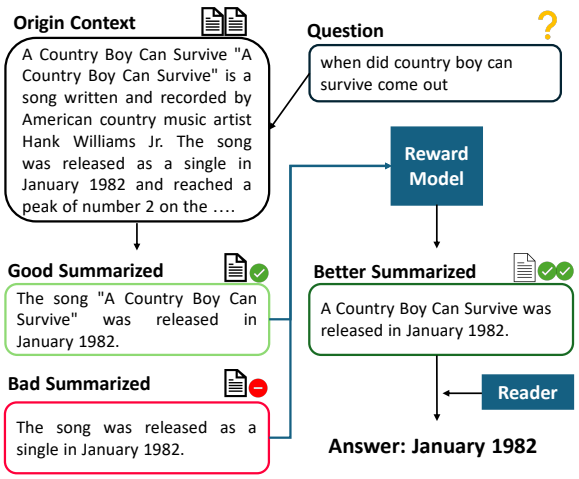

Figure 1: For an effective QA task, we conduct context filtering through the process of creating better summarization using a reward model. Simultaneously, we make it possible to discern which parts are helpful and which are filtered out by utilizing rewards extracted from the data.

1 Introduction

In the rapidly evolving field of NLP, the ability of language models to understand and process long texts has become increasingly important, especially 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). The challenge often lies in distinguishing the useful information from the less relevant details within these texts. 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 by inducing positive and negative feedback using datasets comprised of "chosen" and "rejected" items, with supervised fine-tuning serving as the reward model. We suggest employing this technique for filtering context in QA tasks. We particularly focus on the process of creating 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, thereby contributing to the development of an efficient context filtering model. Through this process, the method ultimately aims to enhance the efficiency of QA models by identifying and retaining only the most relevant information, thereby improving their performance.

As a final point in our experiment summary, we introduce the EM Per Token (EPT) metric and use it

for comparisons between models. This allows us to evaluate which model is more efficient in terms of context length by comparing the number of tokens to the core metric for QA models, the EM score. Additionally, this analysis helps us understand the relationship between context length related to filtering and accuracy metrics in QA tasks.

2 Backgrounds

2.1 Knowledge Refinement

In the task of QA, research focused on retrieving the correct context for a question has been extensively conducted for some time(Chen et al., 2017; Karpukhin et al., 2020; De Cao et al., 2021). Simultaneously, there has been considerable research leveraging the power of LMs to introduce variations such as augmenting question information(Chen et al., 2023) context generation(Yu et al., 2022), summarization(Xu et al., 2023) and paraphrasing(Lee et al., 2023). However, recent studies assert that the presence of irrelevant information within the context can lead to a decrease in the performance of the model(Shi et al., 2023; Akimoto et al., 2023). Therefore, there is a need for research on models that can filter out irrelevant content from the given context while aggregating only the essential information.

2.2 Reward Modeling

With the advancement of LLMs, various methodologies have emerged for training these models to shift away from generating outputs based on toxic or unhelpful datasets. Among the various techniques, Reinforcement Learning from Human Feedback (RLHF) stands out as a significant method, as outlined by Ouyang et al. (2022). This approach involves incorporating human evaluations in specific contexts and suggests improving models via a reward system that learns from this feedback. However, a notable drawback of this approach is the need for labeling to identify human preferences, which can be costly. Additionally, the potential for noise and inconsistencies inherent in human labeling has been highlighted as a disadvantage. To overcome these disadvantages, Rafailov et al. (2023) proposed Direct Preference Optimization(DPO) which leverages Supervised Fine Tuning(SFT) model as a reward model, inducing avoid rejected data and select chosen data from dataset pairs while training. Let y_w is preferred output and y_l is rejected output with given task prompt x and denote Dataset $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^{N}$. We can make loss $\mathcal{L}_R(r_\phi, D)$ with 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

In this experiment, to assess the effectiveness of the filtering process and the reader's ability regarding it, we adopt a *summarize-then-read* pipeline structure with adapting and modifying the structure inspired by Inoue et al. (2021)'s work. During the summarize phase, we compare the SFT and DPO models to determine how efficiently each model summarizes. In the read phase, we compare the ability to find the correct answers within the context that has been filtered out through summarization.

We use XL version of the FLAN-T5(Chung et al., 2022)[1] model from Hugging Face(Wolf et al., 2020) having 3B parameters throughout this experimental design. This seq2seq encoder-decoder model is specialized for summarization to capture the main point of the context. All specific settings, such as prompts and model templates used in the experiments, can be found in the Appendix B.

3.1 Data Generation

In this Experiments, we utilize SQuAD v1.1 (Rajpurkar et al., 2016), Natural Question (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), especially, we gather relevant contexts from DPR(Karpukhin et al., 2020) repository[2] for NQ and TQA. In cases like 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 training and validation dataset, and use the existing validation set as the test set. The datasets postfixed

[1] https://huggingface.co/google/flan-t5-xl
[2] https://github.com/facebookresearch/DPR

Dataset	NQ_r	TQA_r	$SQuAD_r$
Model		**IRA**	
Origin	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 1: This is the ratio that indicates whether the target answer we are seeking is included within the context summarized by each model when the original context is summarized.

with "r" in the table are those that have undergone this process. Detailed information is in Appendix C.

QA task aims to find the corresponding question Q and answer A given a context C. Therefore, we can establish three strategies for summarization through the missing combinations in the necessary information denoted as $I = (Q, A, C)$. We can construct prompts for generating data by dividing them into each type. In the first type, Type 1, it consists of $I_1 = (Q, A, C)$, containing all information, aiming to achieve the best possible summarization. In Type 2, it 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. In Type 3, it 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 Training

Supervised Fine Tuning(SFT) aims to fine-tune the base model with outputs O_1 created with the Type 1 prompt when given a context and question as inputs. The default prompt used for this purpose employs the Type 2 prompt, which has the objective of summarizing the context without answer information, as a prefix. Through this process, the fine-tuned summarization model π^{sft} should be able to utilize all the information included in $I = (Q, A, C)$ to produce the best summarization results.

3.3 DPO Training

For the DPO Training, we need pairs of answer (y_1, y_2) and determine which y would be y_w and y_l satisfying $y_w \succ y_l \mid x$ (Rafailov et al., 2023). Typically human labelers or LLM determine y_w and y_l, but in this architecture scenario, we assume that output 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 utilize Hugging Face TRL framework(von Werra et al., 2020) for running DPO Training.

3.4 Reader Training

We aim to investigate whether an actual reader, when trained with the context filtered by each model, can effectively find answers despite the reduced context length. We conduct reader training to assess the ability to provide answers using the filtered context, which is achieved by summarizing each dataset with the trained summarization models. In other words, we seek to 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 ability of a reader to find answers within a summarized context, we utilize the Exact Match (EM) score and unigram F1 metrics(Rajpurkar et al., 2016). In cases like NQ and TQA, 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 the EM Per Token (EPT) metric what we propose through this study, which serves as an indicator of the efficiency of the given context.

Furthermore, to evaluate the performance of the summarization itself, we introduced a metric called Inclusion Rate of Answer (IRA). This metric assesses how well the summarization is done in terms of leaving room for the reader to find the answer, essentially checking the potential for finding the answer before the reader phase. This can be considered a metric that evaluates the summarization process itself, independent of the subsequent reading phase.

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 2: The results are obtained after training the reader with the context filtered by each model and then evaluating it on the test set. The tokenizer used to calculate mean token length is the same as the one used in the training model.

4 Impact of Context Filtering

Trade-off between Token Length and Accuracy Metrics. In this experiment, all the datasets used are categorized under extractive QA, indicating that to find an answer within a context, it is crucial not only to identify the answer and its relevant context but also to understand the relationship between the contexts that constitute the content relevant to the answer. Through the comparison of the IRA in table 1 and the EM score in table 2, we discover that the length of the context does not necessarily correlate directly with accuracy. Consequently, we examine the contribution of each token to the EM score using the previously defined EPT metric. Specifically, by analyzing the EPT metric with the SFT and DPO models, it was found that in NQ, reducing the token length to 20% still retains 92% of the original performance. In TQA, a token length of 12% maintains 80% performance, and for SQuAD, 12% of the token length achieves 81% of the performance.

From another perspective, in the aforementioned experiment, 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 model remains, indicating room for improvement. Secondly, our approach to filtering the context did not simply involve lexically cutting off existing content or directly borrowing it(Zhiruo Wang, 2023); rather, we restructured it through a prompt, transforming it into a summarization task. This process could account for the observed phenomenon.

Despite this, the results in Table 2 still reveal that the trade-off between token length and accuracy metrics like EM and F1 is favorable; the efficiency gained from filtering outweighs the slight loss in accuracy, indicating that filtering is meaningful and beneficial.

Comparison with Different Reward Model Strategies. During the 3.1 Data Generation phase, we observe that Type 2 outputs, O_2, are summarized in a form that provides short answers about the context and question, while Type 3 outputs, O_3, are more reflective of the lexical elements surrounding the answer in the context. Furthermore, it is confirmed that 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 DPO_{O_1,O_3} focuses on generating shorter outputs centered around elements present in Type 1 but absent in Type 3 outputs, indicating the intended direction of the training results.

5 Conclusion and Future Works

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 than using the 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.

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.

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.

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.

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.

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.

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.

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`.

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.

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*.

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

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}
```

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: These are the specific numbers of data involved in the entire pipeline. For NQ and TQA, the numbers in parentheses indicate the actual amount of data involved.

C Details of Datasets

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