Robust in-context RALM:
simulating noisy contexts resolve noisy retrieval

Anonymous ACL submission

Abstract

Retrieval Augmented Language Models (RALMs) have emerged as a leading approach in Open-Domain Question Answering (ODQA), leveraging external knowledge to enhance answer generation. However, RALMs face challenges when confronted with irrelevant or distracting contexts, particularly in real-world applications with less curated data sources. Addressing these challenges is crucial for improving model accuracy and trustworthiness. In this study, we introduce an innovative in-context learning method Simluate-The-Noise (STN) designed to increase language model resilience in scenarios with absent answers or high distraction. By integrating perturbation techniques with in-context learning, we develop examples that simulate noisy retrieval conditions. Our method notably enhances model robustness without additional training or annotation, enabling the model to accurately identify 'unanswerable' situations in distracting contexts. This cost-effective approach, which simply adds pre-constructed examples to prompts during inference, significantly improves model inference robustness in complex real-world scenarios, thus advancing the reliability of RALMs in ODQA tasks.

1 Introduction

Retrieval Augmented Language Models (RALMs) has become the predominant approach in the domain of Open-Domain Question Answering (ODQA). RALM harness external knowledge to construct answers, thereby enhancing the model's capability to respond to queries beyond data previously trained and improve performance. A typical RALM operates through a two-stage approach: initially retrieving relevant contexts to question and subsequently generating responses based on this retrieved information. Previous studies have validated this as an effective strategy. (Guu et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021; Borgeaud et al., 2022; Ram et al., 2023; Shi et al., 2023b)

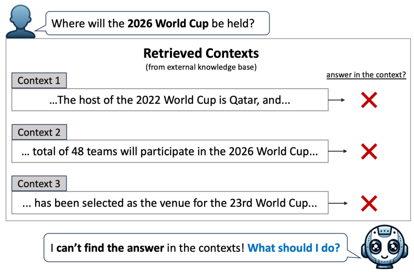

Figure 1: An example of a situation where the retrieved contexts do not contain the answer to a question.

Nonetheless, RALM encounters challenges when the retrieved contexts do not contain information pertinent to the correct answer or when these contexts are filled with distracting elements that could mislead the answer generation process. In real-world applications, these situations are notably more likely to occur as data are gathered from search engines or corporate knowledge bases where the information may not be as reliably organized or accurate as Wikipedia articles. Improving the model's accuracy and apt responses in cases of noisy context is crucial and complex, directly affecting RALM's trustworthiness and robustness. Prior research has primarily managed these challenges by further refining the retrieved context (Nogueira and Cho, 2019; Yu et al., 2022; Glass et al., 2022; Weston and Sukhbaatar, 2023), nonetheless, at the risk of irrelevant contexts influencing the generation phase.

In our study, we present a novel in-context learning method Simulate-The-Noise (STN) to boost language model resilience, effective in both answer-lacking and distraction-rich scenarios. STN uses well-crafted examples to enable the model to identify *unanswerable* cases, enhancing its response accuracy in various contexts.

We develop examples simulating noisy retrieval

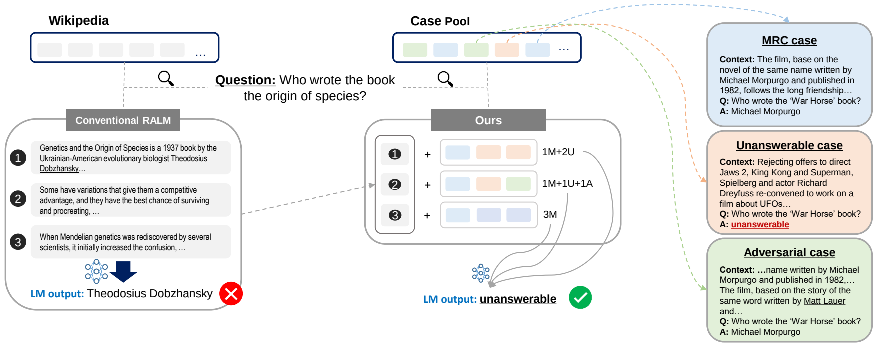

Figure 2: Overview of our approach. Unlike the conventional RALM method, we retrieve cases from the case pool based on the question, and then concatenate these cases with the retrieved contexts to generate the output. This enables more robust inference in noisy retrieval situations (where the correct answer is absent).

conditions, combining in-context learning with perturbation techniques. Our findings reveal that adding the right examples significantly increases model robustness, eliminating the need for extra training or annotation. These examples also help the model to reliably respond with "unanswerable" in distracting contexts, avoiding incorrect answers.

STN is efficient, involving the addition of pre-constructed examples to prompts during inference. It avoids the need for multiple forward passes or auxiliary models, making it cost-effective. This approach holds promise for improving robustness in real-world situations with complex retrieval contexts, thereby enhancing their reliability.

2 Method

2.1 In-context RALM with cases

We focus on enhancing language models (LMs) to identify responses as "unanswerable" in noisy retrieval scenarios, both when the context lacks the correct answer and when it contains only distracting information without the actual answer. Our Retrieval Augmented Language Model (RALM) follows in-context RALM framework (Ram et al., 2023), with a particular focus on Open Domain Question Answering (ODQA) scenarios.

In in-context RALM, for given a question x and answer y, we retrieve documents from external knowledge source and use the k highest ranked documents $d = [d_1, d_2, \ldots, d_k]$. We then concatenate x with d to formulate the answer. The process is represented as:

$$p(y|x) = \sum_{i=1}^{n} p(y|d; x_i) \qquad (1)$$

We enhance this process by incorporating in-context examples. These specially crafted example texts, which we will refer to as *cases*, are represented by $C = \{c_1, c_2, \ldots, c_l\}$, then (1) becomes:

$$p(y|x) = \sum_{i=1}^{n} p(y|C, d; x_i) \qquad (2)$$

The cases we create are essential for directing LMs in noisy contexts. These cases, represented by the set C, should meet following specific requirements:

1. They should be similar to the original task.
2. Adding them should not diminish the model's performance on the original task.
3. They ought to facilitate robust inference in situations of noisy retrieval.

2.2 Crafting cases

We will utilize the SQuAD dataset (Rajpurkar et al., 2016) to construct a case set C_i that enables robust inference in noisy retrieval situations for input question x_i. The SQuAD dataset D is an MRC dataset composed of question (q), passage (p) and answer (a) pairs, represented as $D = \{(q_j, p_j, a_j)\}_{j=1}^{m}$. To this dataset, we will apply a perturbation operation P to create cases similarly structured as (q, p, a) pairs.

QA case To enhance the reasoning capabilities of LMs in ODQA, we supply MRC data as QA cases. Since ODQA essentially involves a reading comprehension task with multiple passages, we use the SQuAD dataset directly without perturbation.

Unanswerable case We craft unanswerable cases to simulate scenarios where the retrieved contexts do not hold answers. In these unanswerable cases, $\tilde{p_j}$ is related to q_j but does not contain the answer. By adding such cases to the prompt, we enable the LM to robustly classify responses as *unanswerable* when the retrieved contexts lack answers.

Prompt	NQ EM	NQ EM (unans)	NQ EM (ans)	NQ F1	TriviaQA EM	TriviaQA EM (unans)	TriviaQA EM (ans)	TriviaQA F1
Baseline	20.33	19.61	20.78	30.46	56.17	25.18	69.46	63.84
1Q	29.09	10.53	40.39	37.88	58.00	10.34	78.44	64.02
3Q	32.96	11.49	46.05	41.74	59.03	8.83	80.56	64.36
2Q+1U	41.61(+8.65)	35.04(+23.55)	45.61(-0.44)	50.14(+8.4)	64.60(+5.57)	31.63(+22.8)	78.74(-1.82)	69.70(+5.34)
5Q	34.54	13.53	**47.35**	42.97	59.19	8.57	**80.90**	64.46
3Q+2U	**44.16**(+9.62)	**40.45**(+26.92)	46.41(-0.94)	**52.21**(+9.24)	**65.98**(+6.79)	**37.01**(+28.44)	78.40(-2.5)	**71.02**(+6.56)

Table 1: Overall performance on the unanswerable datasets. EM (unans) means unanswerable EM and EM (ans) means answerable EM. "Q" represents QA cases, "U" denotes Unanswerable cases, and "A" stands for Adversarial cases. The numbers in parentheses indicate the relative performance improvement of the combined cases compared to the same number of QA cases. The best performance in each column is highlighted in bold.

Prompt	EM	EM (unans)	EM (ans)	F1	EM (unans-only)	EM (adv-unans)
Baseline	19.19	16.05	24.49	25.29	19.30	11.72
1Q	21.46	8.24	43.78	26.79	10.34	5.45
3Q	24.29	9.08	49.96	29.41	11.04	6.48
5Q	25.70	10.71	51.00	30.69	13.43	7.09
2Q+1U	36.92	30.17	48.32	42.10	34.98	23.76
1Q+1U+1A	**41.96**(+5.04)	**37.62**(+7.45)	**49.29**(+0.97)	**46.75**(+4.65)	**43.24**(+8.26)	**30.14**(+6.38)
3Q+2U	40.58	34.80	**50.33**	45.36	40.23	27.57
1Q+2U+2A	**43.96**(+3.38)	**40.36**(+5.56)	50.03(-0.30)	**48.58**(+3.22)	**46.25**(+6.02)	**32.51**(+4.94)

Table 2: Overall performance on the adversarial-unanswerable NQ dataset. EM (unans-only) refers to the Exact Match measured on unanswerable data where the retrieved contexts do not contain adversarial content, while EM (adv-unans) is the Exact Match for data that includes adversarial contexts. Similarly, the best performance in each column is highlighted in bold.

For these cases, we select a passage $\tilde{p_j}$ from D by considering the weighted average of the similarities between $\tilde{p_j}$ and original passage p_j, as well as between $\tilde{p_j}$ and the question q_j, ensuring that $\tilde{p_j}$ doesn't contain the answer a_j. Then we substitute p_j with $\tilde{p_j}$, simulating the noisy retrieval conditions of ODQA using a dense retriever based on the input question. Additionally, we modify the original answer a_j to unanswerable reflecting situations where the relevant information is absent.

Adversarial case We make adversarial cases following the TASA framework (Cao et al., 2022). In TASA, adversarial sentences are created for MRC task by substituting the subjects/objects in the sentence that contain answers (answer sentence) with different entities/nouns. However, in ODQA, the retrieved context often comprises multiple sentences. Thus, instead of adding a single adversarial sentence to the end of the passage, we create and integrate an adversarial passage.

The process of crafting an adversarial passage is as follows:

1. **Rewrite passage:** We use GPT-3.5 to rewrite the original passage p_j, generating a new passage $\hat{p}_j$ that preserves the meaning and answer.
2. **Entity/Noun Substitution:** We make a adversarial sentence using answer sentence in p_j following TASA. Unlike TASA, which substitutes the subject/object with random entities/nouns, we use word vector similarity to find highly similar replacements, thus maintaining the original passage's meaning and embedding similarity. This adversarial sentence then replaces the answer sentence in $\hat{p}_j$.
3. **Truncate and Concatenate:** Since directly combining the p_j and $\hat{p}_j$ would double the length of the passage and introduce redundancy, we truncate them to an appropriate length, and concatenate the $\hat{p}_j$ to p_j.

This approach ensures that our adversarial passages closely mirror the original context while incorporating subtle, challenging variations in order to enhance the robustness of the model in complex ODQA scenarios.

2.3 Case retrieval

Using the aforementioned methods to apply perturbations to the dataset D, we generate a separate case set for each type of perturbation. To utilize the case most similar to the input question x at inference time, we employed a case-based reasoning approach (Thai et al., 2023).

3 Experiment

3.1 Dataset and augmentation

We conducted experiments using two benchmark datasets in ODQA: Natural Questions

(Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). For TriviaQA, we randomly sampled a third of the entire dataset. We selected the Contriever (Izacard et al., 2022a) as our retriever model and used top 5 retrieved contexts for retrieval augmentation. The detailed dataset statistics are in appendix A. We augmented the original NQ and TriviaQA in two distinct ways to create scenarios simulating noisy contexts. Using this dataset, we aim to evaluate how effectively language models (LMs) can respond in situations where the answer is not present in retrieved contexts.

Unanswerable dataset If none of those top-5 retrieved contexts contained the answer string, we replaced the original answer with *unanswerable*.

Adversarial-unanswerable dataset To simulate more challenging and realistic scenarios, we applied adversarial attacks to contexts containing the correct answer. The method of adversarial attacks was the same as that used for case generation, with the difference that we used only the adversarial passage instead of concatenating it with the original. Hence, the adversarial passage also did not contain the answer. Similar to the unanswerable dataset, if the top 5 contexts do not contain the answer, the original answer was changed to *unanswerable*. The aim of this dataset is to measure the model's ability to robustly identify *unanswerable* amidst confusing adversarial information. The detailed statics of augmented datasets are also in appendix A.

3.2 Baseline methods

Language Model In this experiment, the GPT-3.5-turbo-instruct model was employed. We used greedy decoding and kept the seed value fixed throughout the experiment for reproducible results.

Baseline For the baseline, we chose a zero-shot setting that was provided only with instructions to answer with *unanswerable* when an answer could not be found in the contexts.

Baseline with cases To assess the effectiveness of the cases we created, we conducted comparative experiments by adding various combinations of cases to the baseline. Initially, we examined the impact of number of QA cases on ODQA. Then, for a fair comparison, we kept the total number of cases constant while varying their combinations in subsequent experiments.

3.3 Evaluation

Exact Match (EM) and F1 scores are reported following previous literature (Izacard et al., 2022b).

3.4 Results

Table 1 presents the results for the unanswerable dataset. We assessed Exact Match (EM) both for data labeled as "unanswerable" (unanswerable EM) and for data not labeled as such (answerable EM). Initially, adding QA examples improved the answerable EM, but increasing the examples from 3 to 5 did not result in a significant rise. However, appropriately adding unanswerable cases to the QA cases, compared to models with an equal number of QA cases alone, resulted in a substantial increase in unanswerable EM (23.55 and 26.92 in NQ) without decreasing the answerable EM. This suggests that our crafted unanswerable cases enhanced the LM's ability to respond with 'unanswerable' in noisy context situations.

Table 2 shows the results for the adversarial-unanswerable dataset, revealing a trend similar to that observed in the unanswerable dataset. While increasing the number of QA cases did enhance the answerable EM, using a well-combined set of cases yielded higher EM scores. Notably, in the adversarial unanswerable data, adding adversarial cases proved more effective than using only unanswerable cases. This demonstrates that a strategic combination of cases can significantly enhance the LM's robustness in more complex, noisy situations. This demonstrates that providing well-designed cases appropriately in conjunction with simple in-context examples allows the model to infer robustly in such scenarios.

4 Related Work

We discuss the development of the RALM framework and also introduce previous literature that discussed the robustness of RALMs in the Appendix.

5 Conclusion

In our experiment, we explored how LMs respond in noisy retrieval situations and the impact of the cases we created in such scenarios. We found that simply adding in-context examples (QA cases) is not sufficient to address those. However, when well-designed cases were utilized, there was a significant improvement in performance under noisy retrieval conditions, and it was confirmed that more robust inference is possible even in more complex situations through various combinations. This suggests that our research could provide a new methodology to fully harness the reasoning capabilities of LMs by offering appropriate examples.

6 Limitations and Risk

Our work tries to make the RALM process to be more robust by simulating noisy context settings. One limitation of this approach is that our cases require a reading comprehension dataset. So in case, there is a large domain shift, such as biomedical ODQA, we might not perform so well.

Acknowledgements

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Yu Cao, Dianqi Li, Meng Fang, Tianyi Zhou, Jun Gao, Yibing Zhan, and Dacheng Tao. 2022. TASA: Deceiving question answering models by twin answer sentences attack. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11975–11992, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. *arXiv preprint arXiv:2207.06300*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Édouard 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.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Mandar Joshi, Eunsol Choi, Daniel S 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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yoav Levine, Itay Dalmedigos, Ori Ram, Yoel Zeldes, Daniel Jannai, Dor Muhlgay, Yoni Osin, Opher Lieber, Barak Lenz, Shai Shalev-Shwartz, et al. 2022a. Standing on the shoulders of giant frozen language models. *arXiv preprint arXiv:2204.10019*.

Yoav Levine, Ori Ram, Daniel Jannai, Barak Lenz, Shai Shalev-Shwartz, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022b. Huge frozen language models as readers for open-domain question answering. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

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.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

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.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Datasets	size	IR Recall	unanswerable ratio
NQ	3,610	0.62	0.38
TriviaQA	3,771	0.69	0.31

Table 3: Dataset statistics

Type	size	rate (%)
answerable	1343	0.37
unanswerable-only	1295	0.36
adversarial-unanswerable	972	0.27

Table 4

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Dung Thai, Dhruv Agarwal, Mudit Chaudhary, Wenlong Zhao, Rajarshi Das, Manzil Zaheer, Jay-Yoon Lee, Hannaneh Hajishirzi, and Andrew McCallum. 2023. Machine reading comprehension using case-based reasoning. *arXiv preprint arXiv:2305.14815*.

Orion Weller, Aleem Khan, Nathaniel Weir, Dawn Lawrie, and Benjamin Van Durme. 2022. Defending against poisoning attacks in open-domain question answering. *arXiv preprint arXiv:2212.10002*.

Jason Weston and Sainbayar Sukhbaatar. 2023. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. *arXiv e-prints*, pages arXiv–2303.

A Appendix

A.1 Dataset statistics

Tables 3 and 4 show dataset statistics. Table 3 presents the statistics of the original NQ and TQA. Table 4 shows the statistics for the adversarial-unanswerable dataset of NQ. They include the number and proportion of each data type.

A.2 Case based reasoning

We conducted additional experiments to validate the effectiveness of the case-based reasoning approach. Focusing on an unanswerable dataset, we compared the results of retrieving cases randomly from the entire case set with those obtained using case-based reasoning. Table 5,6 presents these results. Table 5 shows performance on NQ and TriviaQA unanswerable datasets. Table 6 shows the performance of the NQ adversarial-unanswerable dataset. In both datasets, the findings demonstrate that retrieving cases using case-based reasoning is more effective.

A.3 Related works

In-context RALMs Traditionally, Retrieval-Augmented Language Models (RALMs) involved training a separate reader to generate answers based on the retrieved documents (Lewis et al., 2020; Izacard and Grave, 2021). However, it has been recently discovered that large language models can be used as readers without additional training. (Levine et al., 2022b,a) Moreover, it has been shown that enhancing performance is possible either by further training the retriever (Shi et al., 2023b) or simply by concatenating documents to the query (Ram et al., 2023).

Robustness of RALMs RALMs are demonstrating exceptional performance in knowledge-intensive tasks by merging external knowledge with the generative capabilities of language models. Recent studies indicate that large language models are sensitive to the retrieved context, with irrelevant context actually degrading performance. (Longpre et al., 2021; Weller et al., 2022; Shi et al., 2023a) In cases where there are conflicts between retrieved contexts, or when information is absent, several researches utilize prompting (Zhou et al., 2023) or train separate calibrators (Chen et al., 2022) to resolve these issues. Our approach can be described as maximizing the reasoning capabilities of language models (LMs) by using appropriate prompts when information is lacking.

Prompt	NQ EM	NQ EM (unans)	NQ EM (ans)	NQ f1	TriviaQA EM	TriviaQA EM (unans)	TriviaQA EM (ans)	TriviaQA f1
3Q+2U	**44.16**	**40.45**	46.41	**52.21**	**49.28**	**37.01**	78.40	**71.02**
3Q+2U (R)	40.89	33.58	45.34	49.28	64.81	30.48	79.54	69.94

Table 5: Performance on the unanswerable dataset. (R) indicates the results of randomly retrieving cases.

Prompt	**EM**	**EM (unans)**	**EM (ans)**	**EM**	**EM (unans only)**	**EM (adv-unans)**
1Q+2U+2A	43.961	40.362	50.037	48.587	46.255	32.51
1Q+2U+2A (R)	41.053	35.333	50.707	45.77	40.695	28.189

Table 6: Performance of the NQ adversarial-unanswerable dataset. (R) indicates the results of randomly retrieving cases.