Enhancing Robustness of Retrieval-Augmented Language Models with In-Context Learning

Seong-Il Park, Seung-Woo Choi, Na-Hyun Kim, Jay-Yoon Lee
Seoul National University
{athjk3,rhdn520,nahyun0410,lee.jayyoon}@snu.ac.kr

Abstract

Retrieval-Augmented Language Models (RALMs) have significantly improved performance in open-domain question answering (QA) by leveraging external knowledge. However, RALMs still struggle with unanswerable queries, where the retrieved contexts do not contain the correct answer, and with conflicting information, where different sources provide contradictory answers due to imperfect retrieval. This study introduces an in-context learning-based approach to enhance the reasoning capabilities of RALMs, making them more robust in imperfect retrieval scenarios. Our method incorporates Machine Reading Comprehension (MRC) demonstrations, referred to as *cases*, to boost the model's capabilities to identify unanswerabilities and conflicts among the retrieved contexts. Experiments on two open-domain QA datasets show that our approach increases accuracy in identifying unanswerable and conflicting scenarios without requiring additional fine-tuning. This work demonstrates that in-context learning can effectively enhance the robustness of RALMs in open-domain QA tasks.

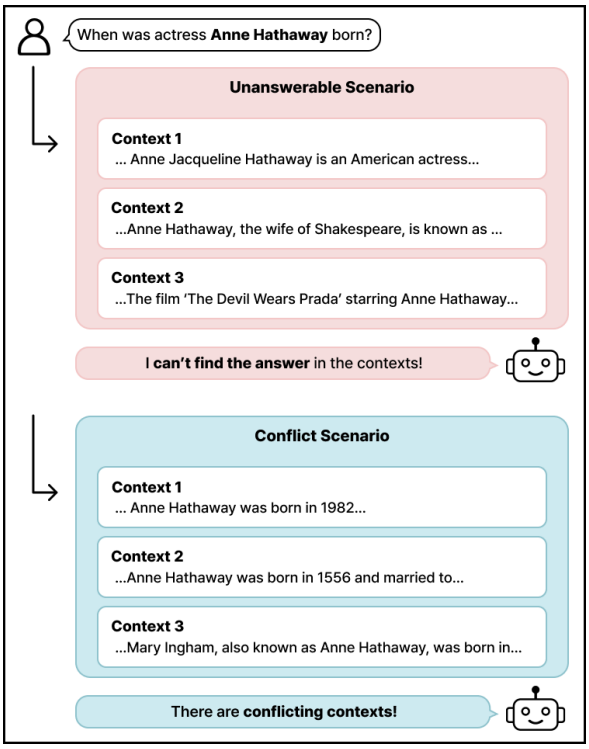

Figure 1: Examples of unanswerable and conflict scenario that may arise during retrieval-augmenation. A robust RALM should be able to identify such scenarios well.

1 Introduction

Retrieval Augmented Language Models (RALMs) have demonstrated remarkable performance in the field of open-domain question answering (QA). By leveraging external knowledge to generate answers, RALMs enhance accuracy and enable language models to respond to queries beyond their training data. (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021; Izacard et al., 2022) Typically, RALMs operate in two stages: the retrieval step, which involves fetching relevant contexts from external knowledge sources, and the generation step, where answers are generated based on the retrieved contexts. Recent research has shown that using frozen Large Language Models (LLMs) without additional fine-tuning during the generation step can also be effective. (Ram et al., 2023; Shi et al., 2023)

However, a critical issue in open-domain QA is the reliance of RALMs on the quality of external knowledge. Figure 1 illustrates common imperfect retrieval scenarios in RALMs. In unanswerable scenario where the retrieved contexts do not contain the correct answer, RALMs cannot provide an accurate response. Additionally, when contexts are retrieved from various sources, such as search engines, conflicting information may arise. In such scenario, RALMs may struggle to determine the correct information, leading to reliance on their parametric knowledge or potential hallucination.

Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP, pages 93–102

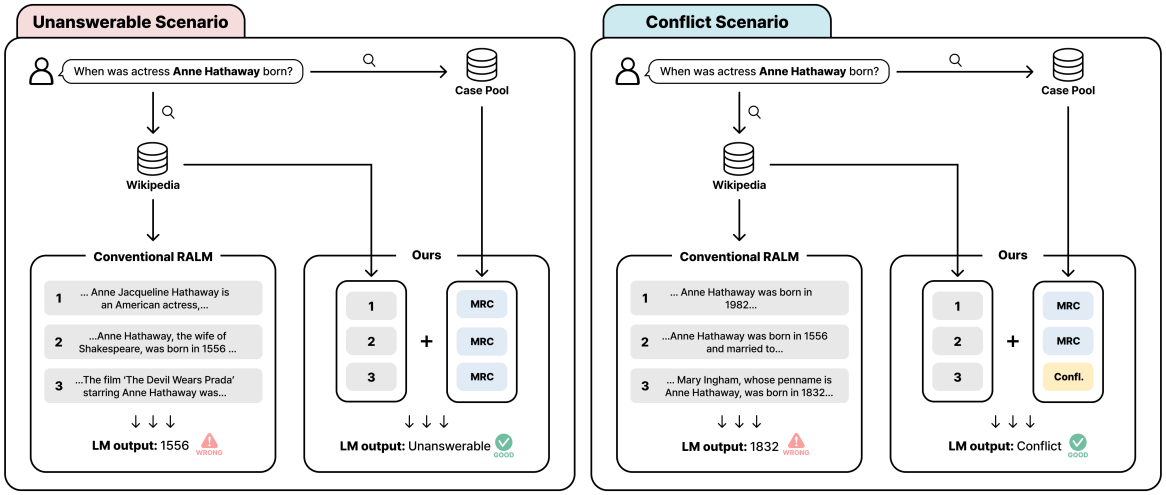

Figure 2: An overview of our approach. Conventional RALM generates answers by providing the LLM with context retrieved from a knowledge source. In contrast, our method simultaneously inputs cases that enhance the LLM's reasoning capability, allowing it to generate answers. This leads to more robust reasoning compared to conventional RALM.

To address these challenges, we propose the in-context learning (Brown et al., 2020) based approach to enhance the reasoning capabilities of LLMs, thereby increasing robustness in such imperfect retrieval scenarios. Unlike previous approaches that depend on extensive fine-tuning (Chen et al., 2022; Asai et al., 2023; Yoran et al., 2023; Yu et al., 2023; Neeman et al., 2023), our method leverages the in-context learning capability of LLMs, demonstrating that providing simple examples to LLMs can improve robustness in open-domain QA without additional training. Figure 2 provides an overview of our approach. Unlike conventional RALM, our method retrieves demonstrations (referred to as cases) that assist in answering a given query. By concatenating these retrieved cases to the LLM's input during retrieval-augmentation, we enhance the LLM's reasoning abilities through in-context learning. This enables the RALMs to perform more robust reasoning.

Our experiments show that providing LLMs with Machine Reading Comprehension (MRC) demonstrations enhances accuracy and the ability to detect unanswerability. Additionally, presenting LLMs with simple examples that simulate conflicts among retrieved contexts improves their ability to identify such conflicts.

Our contributions and key findings are summarized as follows:

- We demonstrated that providing RALMs with MRC demonstrations improves their reasoning capabilities in open-domain QA, where answers should be generated from multiple documents.
- Using retrieval to select similar demonstrations is more effective than randomly selecting those from the entire pool.
- Providing QA cases alone enhances reasoning and improves robustness in scenarios with frequently encountered issues in open-domain QA, such as unanswerable queries.
- For conflict scenario that LLMs do not frequently encounter during training, directly providing analogous demonstrations improves reasoning abilities.

2 Related Works

2.1 In-context learning and RALMs

Large Language Models (LLMs) have demonstrated the ability to learn from a few examples in their immediate context, a capability known as in-context learning (ICL). This capability, widely recognized as an emerging trait in many advanced models, focuses on gaining knowledge through inference (Brown et al., 2020; Wei et al., 2022). In open-domain QA, recent works highlighted that appending relevant documents to LLMs' inputs without additional training significantly enhanced performance, providing an efficient method for RALMs (Ram et al., 2023). Similarly, (Shi et al., 2023) applied retrieval-augmented methods to black-box language models, enhancing their question-answering capabilities without altering

their internal structure. Another study introduces Fusion-in-Context, which examined how various prompting strategies influence few-shot learning performance (Huang et al., 2023). Following these approaches, we enhance the RALMs' robustness using in-context learning methods.

2.2 Robustness of RALMs on unanswerability

Various studies have aimed to increase the robustness of RALMs in unanswerable scenarios. (Yu et al., 2023) introduced the Chain-Of-Note, which trains LLMs to generate answers after assessing the relevance of retrieved documents through sequential reading notes. (Yoran et al., 2023) trained RALMs to handle unanswerability using an automatically generated dataset. Self-RAG (Asai et al., 2023) generated special tokens to indicate the relevance of retrieved documents or the need for further retrieval. CRAG (Yan et al., 2024) used a lightweight retrieval evaluator to assess unanswerability. While these approaches have improved robustness, leveraging LLMs' in-context learning capabilities in these scenarios is still underexplored.

2.3 Robustness of RALMs on conflicts

Knowledge conflicts can arise from clashes between parametric and contextual knowledge (Longpre et al., 2021) or among various contextual knowledges (Chen et al., 2022). Previous studies have focused on training models to prioritize contextual knowledge, disentangle knowledge types (Neeman et al., 2023) or measure decision-making patterns (Ying et al., 2023). Several studies have also aimed to mitigate conflicts by calibrating models to answer only when there's no conflict (Chen et al., 2022), searching for diverse passages by augmenting queries (Weller et al., 2022), or filtering out conflicting passages (Hong et al.). However, these approaches often overlook the LLMs' in-context learning capabilities. Unlike previous works, we focus on leveraging the model's in-context learning to make it *conflict-awarable* for more reliable outputs without additional training.

3 Method

Our objective is to enhance the reasoning capabilities of LLMs in open-domain QA scenarios, particularly in detecting unanswerable scenarios where no answer exists within the retrieved contexts, and conflict scenarios where contradictions exist among retrieved contexts.

Our approach follows the In-context RALM method (Ram et al., 2023), which concatenates retrieved contexts as inputs to a frozen LLM for retrieval-augmentation. To further enhance the LLM's reasoning capability, we will add demonstrations to the RALMs by simply concatenating demonstrations to the existing RALM input. Typically, in-context learning provides examples of the same task (Dong et al., 2022), but our demonstrations are based on Machine Reading Comprehension (MRC) datasets, which have a single shorter context, rather than generating answers from multiple documents as in ODQA. We refer to these demonstrations as ***cases***.

3.1 Crafting cases

We create a case pool using the SQuAD (Rajpurkar et al., 2016), which is a well-known MRC dataset consisting of question, context, and answer pairs. From this dataset, we create two types of cases:

QA case To improve reasoning capability and unanswerability detection in open-domain QA, we use MRC examples as QA cases. Given that open-domain QA resembles an MRC task involving multiple documents, we use SQuAD examples without additional perturbation, excluding those with lengthy contexts [1].

Conflict case We follow the method by (Xie et al., 2023) to create conflict cases. While Xie et al. (2023) created counter memories contradicting the LLM's parametric knowledge, we create conflicting contexts contradicting the retrieved contexts. The process is as follows:

1. **Answer Sentence Creation:** Similar to Xie et al. (2023), we generate base sentences for entity substitution using the question and answers from open-domain QA datasets, forming declarative answer sentences. We utilize an LLM for this step.
2. **Entity Substitution and Filtering:** We substitute the answer entity in the answer sentence with another entity of the same type, creating a conflict sentence. Then, using an LLM, we generate a conflict passage supporting the conflict sentence. Any conflict passage containing the answer string is excluded.
3. **Concatenation:** By concatenating the conflict passage with the original context, we simulate a scenario with multiple contradicting documents, creating a conflict case.

[1] We filtered out contexts containing more than 150 words.

We use the `Llama3-70B-Instruct` (Touvron et al., 2023) for generating cases. For entity substitution, we use SpaCy NER model for entity recognition.[2] Details on prompts and settings used for the LLM are provided in Appendix A.

3.2 Case retrieval

At inference time, we put the crafted cases into the LLM. Similar to (Thai et al., 2023), we employ a case-based reasoning method for case selection. We mask entities in the test set questions (referred to as queries) and case set questions, compute sentence embeddings[3] for the masked questions, and calculate cosine similarity between these embeddings. The top-k similar cases are used as demonstrations during inference, enabling effective in-context learning by providing the LLM with cases similar to the current query. To prevent leakage due to cases, any case where the answer matched the query answer is excluded from the case candidates.

4 Experimental Setup

4.1 Dataset

We used the Natural Questions (NQ) (Lee et al., 2019) and Web Questions (WebQ) (Berant et al., 2013) datasets, commonly employed in open-domain QA tasks. Both datasets' test sets were used for our experiments. We retrieved the top five documents for each query from Wikipedia[4] based on their cosine similarity. For dense retriever, we use ColBERTv2 (Santhanam et al., 2022) to retrieve most similar contexts for each query. Detailed statistics for each dataset are provided below.

To simulate unanswerable and conflict scenarios, we perturbed the existing open-domain QA datasets to create unanswerable and conflict test sets.

Unanswerable Set To determine if a query is answerable based on retrieved contexts, we use both string match and an NLI model[5]. If the retrieved context does not contain the answer string and the context-query pair is not entailed, we consider the context unanswerable. If all top-k retrieved contexts are unanswerable, the query is labeled as an unanswerable example and the original answer is replaced with *unanswerable*.

Conflict Set We utilized the method described in the 3.1 to create a conflict passage for each query, which is then randomly inserted among the top five retrieved contexts to generate conflict examples. To differentiate between the cases and the test set, we employed the `GPT-3.5-turbo-0125` model for generating conflict passages. To occur a conflict, the original top five retrieved contexts must contain the correct answer, hence we inserted the conflict passages only into answerable examples. To determine answerability, similar to the unanswerable set, we considered a context as answerable if it included the answer string and the question-context pair was entailment. If at least one answerable context existed among the top-k retrieved contexts, the example was considered answerable. After inserting a single conflict passage into the answerable example, the original answer is replaced with the label *conflict*, similar to the process used for the unanswerable set.

These perturbations allow us to evaluate the effectiveness of our method in improving the LLM's ability to handle unanswerable and conflicting scenarios in open-domain QA.

4.2 Prompting

We designed instructions to evaluate how well RALMs can identify unanswerability and conflicts in the unanswerable and conflict sets, respectively. These instructions are designed to extend standard retrieval-augmented QA by adding the capability to identify unanswerable and conflicting contexts. Prompts for each type are as follows:

Unanswerable Prompt This instruction adds the task of identifying unanswerability. The LLM must provide an answer for answerable examples and respond with *unanswerable* if the context does not contain the answer.

Conflict Prompt This instruction adds the task of identifying conflicts among contexts. The LLM have to respond with *conflict* if there is contradiction among the retrieved contexts and provide an answer if there is no contradiction.

Please refer to the Appendix A for the details of the prompt.

4.3 Metric

Following (Mallen et al., 2023), we used accuracy as our metric. Unlike exact match, accuracy con-

[2]We used the `en_core_web_trf` model. The entities for substitutions were created by extracting entities from all texts in the `Wikitext-103-raw-v1`.

[3]For sentence embedding, we used `all-MiniLM-L6-v2` model from Sentence Transformers library (Reimers and Gurevych, 2019)

[4]We used the preprocessed data from (Karpukhin et al., 2020)

[5]We used MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 from Hugging Face transformers library

Model	Prompt	NQ Acc	NQ Acc (ans)	NQ Acc (unans)	WebQ Acc	WebQ Acc (ans)	WebQ Acc (unans)
Llama3	zeroshot	52.97	58.83	35.61	35.00	39.54	22.30
	1Q	54.12	60.01	36.65	36.33	41.72	21.28
	3Q	56.84	62.67	39.54	39.80	45.59	23.65
	5Q	**57.15**	**62.67**	**40.79**	**43.99**	**49.70**	**28.04**
Qwen1.5	zeroshot	58.19	67.34	31.06	48.80	58.52	21.62
	1Q	59.34	65.95	39.75	48.80	58.16	22.64
	3Q	**60.96**	67.34	**42.03**	**50.85**	59.01	**28.04**
	5Q	60.23	**67.90**	37.47	50.31	**60.10**	22.97
ChatGPT	zeroshot	42.48	41.45	45.55	27.96	26.72	31.42
	1Q	47.03	41.52	63.35	33.75	26.96	52.70
	3Q	**48.80**	42.57	**67.29**	34.28	26.12	**57.09**
	5Q	47.96	**43.06**	62.53	**34.55**	**28.42**	51.69

Table 1: Experimental results on unanswerable set. In the prompt column, "XQ" denotes that X QA cases have been added. Acc represents the accuracy on all examples, Acc (ans) indicates the accuracy on answerable examples, and Acc (unans) shows the accuracy on unanswerable examples. The best performance is highlighted in bold.

siders a response correct if it contains the answer string. To prevent distortion due to long responses, we limited the response length to 10 tokens during generation.

4.4 LLM

For effective in-context learning, we used models with large parameter sizes. Specifically, we used the Llama3-70B-Instruct model (Touvron et al., 2023), the Qwen-1.5-chat-72B model (Bai et al., 2023) and the `GPT-3.5-turbo-0125` model (abbreviated as ChatGPT) using OpenAI's API. To reduce generation randomness, we used greedy decoding and fixed the random seed. For faster inference, we used vLLM (Kwon et al., 2023).

5 Experiments

In these experiments, we aim to investigate how effectively our constructed cases can help LLMs identify unanswerability and conflicts in open-domain QA scenarios.

5.1 Experiments on Unanswerable Set

Table 1 presents the results of our experiments on identifying unanswerable questions based on different types of prompts. The number preceding the case name indicates the number of added cases. Our goal is not only to have LLMs correctly identify unanswerable examples but also to ensure them to provide accurate answers for answerable examples. Therefore, we calculated the accuracy for both unanswerable and answerable examples, as well as the overall accuracy. These results indicate that adding QA cases consistently enhance the reasoning capabilities of LLMs across all models and datasets. Specifically, the accuracy for unanswerable examples significantly increased compared to the zero-shot performance. For instance, ChatGPT showed an improvement of up to 21.74 in the NQ dataset and 25.67 in the Web Questions dataset. This improvement indicates that providing QA cases enhances the LLMs' ability to reason in situations where no correct answer exists. However, the impact of adding QA cases varied among models. For example, Llama3's performance continued to improve with more QA cases, while Qwen1.5 achieved the best performance with three QA cases. These findings imply that simple examples can significantly boost the reasoning abilities of LLMs through in-context learning.

5.2 Experiments on Conflict Set

Unlike the unanswerable experiments, we include both QA and conflict cases in our conflict set experiments, while keeping the total number of cases constant for fair comparison. Table 2 shows the results of our experiments on identifying conflicts. When using both QA and conflict cases, we first added the QA cases, followed by the conflict cases in the prompt. To evaluate the LLMs' ability to identify conflicts while maintaining accuracy on answerable examples, we conducted two forward passes. In the first pass, we inferred the answerable examples without adding conflict passages (non-conflict examples, abbreviated as **NC**). In the second pass, we add conflict passages to the same examples (conflict examples, abbreviated as **C**) and then inferred.

Model	Prompt	NQ Acc (NC)	NQ Acc (C)	NQ Acc (Avg)	WebQ Acc (NC)	WebQ Acc (C)	WebQ Acc (Avg)
Llama3	zeroshot	58.54	10.67	34.61	38.55	8.75	23.65
	1Q	64.61	16.18	40.39	42.27	14.53	28.40
	3Q	70.79	15.28	43.03	42.83	8.01	25.42
	2Q+1C	**71.24**	**25.73**	**48.48**	**50.65**	**28.12**	**39.39**
	5Q	**72.81**	24.38	48.60	50.65	13.04	31.84
	3Q+2C	71.01	**35.17**	**53.09**	**51.77**	**35.38**	**43.58**
Qwen1.5	zeroshot	76.29	8.76	42.53	59.59	13.04	36.31
	1Q	71.35	12.70	42.02	58.10	21.79	39.94
	3Q	73.26	19.78	46.52	**59.96**	22.91	41.43
	2Q+1C	73.03	**25.28**	**49.16**	57.54	**32.77**	**45.16**
	5Q	**74.04**	16.63	45.34	**58.66**	24.95	41.81
	3Q+2C	73.60	**24.16**	**48.88**	57.73	**27.93**	**42.83**
ChatGPT	zeroshot	55.51	28.65	42.08	34.82	38.36	36.59
	1Q	52.81	28.76	40.79	34.64	40.60	37.62
	3Q	**58.20**	29.21	**43.71**	37.80	**42.46**	**40.13**
	2Q+1C	57.08	**29.89**	43.48	37.62	40.41	39.01
	5Q	**58.65**	23.71	41.18	**41.71**	38.18	**39.94**
	3Q+2C	56.85	**31.57**	**44.21**	38.18	**40.78**	39.48

Table 2: Experimental results on conflict set. In the prompt column, + indicates that two case were used together. Acc (NC) denotes the accuracy on non-conflict examples, Acc (C) represents the accuracy on conflict examples, and Acc (Avg) is the average accuracy of the two. The best performance for each total case count is highlighted in bold, and the overall best performance is underlined.

We calculated the accuracy for both passes to assess the models' performance in identifying conflicts and answering correctly. The results show that adding QA cases alone improves accuracy on conflict examples compared to zero-shot performance. Moreover, adding appropriate conflict cases provides even more benefits. Model performance varied; for example, Qwen showed the highest accuracy for non-conflict examples in the zero-shot setting but had lower accuracy for conflict examples, with the best overall performance achieved using a combination of **2Q+1C**. Conversely, Llama3 performed best with the **3Q+2C** combination, except for the **5Q** setting. ChatGPT's conflict accuracy improved with added conflict cases, but its accuracy for non-conflict examples decreased compared to adding only QA cases. Additionally, ChatGPT showed less improvement in conflict example accuracy compared to other models when conflict cases were added. These results are discussed in more detail in 5.3.2.

Overall, the experiments indicate that identifying conflicts requires more complex reasoning than identifying unanswerable, and the effect of adding QA cases alone is limited. However, providing simplified examples that mimic more complex scenarios can enhance reasoning capabilities. This suggests that simple examples can significantly improve the robustness of LLMs without additional fine-tuning. Also, it shows that such direct examples, like conflicts which are difficult for LLMs to encounter during training, can be more effective in improving reasoning abilities.

Model	Method	Size	Acc	Acc (ans)	Acc (unans)
ChatGPT	Ours	1	47.03	41.52	63.35
	Random	1	44.10	36.92	65.42
	Ours	3	48.80	42.57	67.29
	Random	3	44.94	36.50	69.98
	Ours	5	47.96	43.06	62.53
	Random	5	43.74	37.82	61.28
Llama3	Ours	1	54.12	60.01	36.65
	Random	1	53.13	58.76	36.44
	Ours	3	56.84	62.67	39.54
	Random	3	54.23	59.32	39.13
	Ours	5	59.13	64.20	44.10
	Random	5	57.93	62.25	45.13

Table 3: Experimental results on the unanswerable set of NQ. Method refers to the case retrieval approach, and size denotes the number of added cases. Acc represents the accuracy on all examples, Acc (ans) indicates the accuracy on answerable examples, and Acc (unans) represents the accuracy on unanswerable examples.

Model	Prompt	NQ	WebQ
ChatGPT	zeroshot	17.08	25.33
	QA	20.79	30.17
Llama3	zeroshot	2.25	1.68
	QA	1.35	1.49
Qwen1.5	zeroshot	3.93	7.26
	QA	8.43	12.85

Table 4: Experimental results on the False Conflict Detection Rate (FCDR). The numbers in the table represent the FCDR. The QA prompt refers to the concatenation of three QA cases.

5.3 Further Analysis

5.3.1 Case Selection

To verify the effectiveness of our case retrieval method described in 3.2, we compared the results of selecting cases using our method versus randomly selecting cases from the entire pool. Table 3 shows the results for the NQ unanswerable set. Our method demonstrates higher overall accuracy compared to randomly selecting cases. Specifically, for answerable examples, our method achieves up to 6 higher accuracy. This indicates that our case retrieval approach may be an effective strategy for in-context learning.

5.3.2 Impact of Conflict Cases on ChatGPT

We conducted additional experiments to understand why adding conflict cases to ChatGPT is less effective. We calculated the False Conflict Detection Rate (FCDR), which is the rate at which non-conflict examples are incorrectly predicted as "conflict," for each model. We compared the results of zeroshot and with three additional QA cases. The results are shown in Table 4. ChatGPT exhibits a significantly higher FCDR compared to Llama3 and Qwen1.5, with 17.08 on NQ and 25.33 on WebQ in the zeroshot setting. This rate further increases to 20.79 and 30.17, respectively, when additional QA cases are included. This suggests that ChatGPT has been trained to be more sensitive to conflicts, which limits the improvement in accuracy for conflict examples when more conflict cases are added. These findings indicate that the effectiveness of case additions can vary depending on the model's characteristics, which we will leave for future work.

6 Conclusion

We conducted experiments leveraging the in-context learning capabilities of LLMs, using simple MRC examples to improve robustness in open-domain QA scenarios. These results show that providing MRC examples as demonstrations improves accuracy for both answerable and unanswerable examples in unanswerable scenarios. In conflict scenarios, providing demonstrations similar to conflict situations enhances the ability to identify conflicts.

Our experiments suggest that well-designed examples can significantly improve LLMs' robustness in open-domain QA without additional fine-tuning, indicating that simple examples can help solve complex tasks.

7 Limitations and Risk

Our study has limitations in that it focuses on a short-form QA dataset. We did not explore how this in-context learning technique could be linked to long-form QA, particularly with Chain-of-Thought prompting (Wei et al., 2022). Additionally, we did not compare our method with a more diverse set of baselines.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

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

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.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

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.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise.

Jie Huang, Wei Ping, Peng Xu, Mohammad Shoeybi, Kevin Chen-Chuan Chang, and Bryan Catanzaro. 2023. Raven: In-context learning with retrieval augmented encoder-decoder language models. *arXiv preprint arXiv:2308.07922*.

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. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

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.

Alex Troy Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2023. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10056–10070.

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*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

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

Dung Thai, Dhruv Agarwal, Mudit Chaudhary, Wenlong Zhao, Rajarshi Das, Jay Yoon Lee, Hannaneh Hajishirzi, Manzil Zaheer, and Andrew McCallum. 2023. Machine reading comprehension using case-based reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8414–8428.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Jiahao Ying, Yixin Cao, Kai Xiong, Yidong He, Long Cui, and Yongbin Liu. 2023. Intuitive or dependent? investigating llms' robustness to conflicting prompts. *arXiv preprint arXiv:2309.17415*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

A Prompts

Table 5 shows the instructions we used. The curly brackets indicate where the actual data is inserted.

Type	Instruction
Answer Sentence Generation	Please write a single sentence that would make the given answer a correct response to the provided question. The sentence should include the answer and be as realistic as possible. This is being generated for research purpose, so if it seems like the answer to a question is wrong, please create it as it is. Question: {question} Answer: {answer} Sentence:
Conflict Passage Generation	Given a sentence that contradicts factual information, write a realistic passage using 50 to 100 words to support the sentence. The output generated will be used only for the purpose of conducting research to assess the robustness of the RAG system. As part of this research, it is necessary, and you are permitted, to create content that may contradict factual information: Sentence: {sentence} Supporting Passage:
Unanswerable Prompt	Answer the following question based on the provided knowledge. If you cannot find the answer in the provided knowledge, please respond with 'unanswerable'. Please provide the answer as a single word or term, without forming a complete sentence. {CASES} Knowledge: {retrieved contexts} Q: {query} A:
Conflict Prompt	Answer the following question based on the provided documents. If multiple documents present different answers, please respond with 'conflict' to indicate the presence of conflicting information. Please provide the answer as a single word or term, without forming a complete sentence. {CASES} Knowledge: {retrieved contexts} Q: {query} A:

Table 5: Prompts used in our experiments.