RE-RAG: Improving Open-Domain QA Performance and Interpretability with Relevance Estimator in Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

The Retrieval Augmented Generation (RAG) framework utilizes a combination of parametric knowledge and external knowledge to demonstrate state-of-the-art performance on open-domain question answering tasks. However, the RAG framework suffers from performance degradation when the query is accompanied by irrelevant contexts. In this work, we propose the RE-RAG framework, which introduces a relevance estimator (RE) that not only provides relative relevance between contexts as previous rerankers did, but also provides confidence, which can be used to classify whether given context is useful for answering the given question. We propose a weakly supervised method for training the RE simply utilizing question-answer data without any labels for correct contexts. We show that RE trained with a small generator (sLM) can not only improve the sLM fine-tuned together with RE but also improve previously unreferenced large language models (LLMs). Furthermore, we investigate new decoding strategies that utilize the proposed confidence measured by RE such as choosing to let the user know that it is "unanswerable" to answer the question given the retrieved contexts or choosing to rely on LLM's parametric knowledge rather than unrelated contexts.

1 Introduction

In recent years, the retrieval augmented generation framework has shown promising progress in natural language generation, specifically on knowledge-intensive tasks. This approach has been studied in many forms, from traditional RAG (Lewis et al., 2020b), which aggregates answers from multiple contexts using document relevance scores as weights, to approaches like RALM (Ram et al., 2023), which simply utilizes concatenated context as an in-context learning approach for large-language models (LLMs). Retrieval augmented generation enhances the model's faithfulness and reliability by leveraging nonparametric knowledge on top of parametric knowledge (Luo et al., 2023). In particular, the RAG framework has the advantage of being easily adaptable to modern LLMs (Brown et al., 2020; Touvron et al., 2023). These advantages have sparked a significant amount of new research (Asai et al., 2023; Lin et al., 2023; Shi et al., 2023) focused on the RAG framework.

Despite the great potential of the retrieval augmented generation framework, if the language model is provided with contexts that are not relevant to the query, it will be distracted by these inappropriate contexts, negatively affecting the accuracy of the answers (Yoran et al., 2023). While retrievers or re-rankers in existing research have been effective at measuring the relative ranking across contexts to a query, these modules often fail to determine whether top-ranked contexts are actually relevant to the query or not. Furthermore, if a precise relevance score is not used in the traditional RAG framework, it can cause problems such as directing attention to documents that are less likely to answer the query.

In this work, we propose the `RE-RAG` framework, which extends traditional RAG by incorporating a relevance estimator (`RE`) to simultaneously measure the precise relative relevance between retrieved contexts and evaluate their confidence, which can be used to classify whether given context is useful for answering the given question. By more accurately measuring the relative relevance between contexts, RE computes precise relevance scores suitable for weighted aggregated answers in the traditional RAG framework and also acts as an efficient reranker. RE trained on a small generator (sLM) not only benefits sLM fine-tuned together with RE but can also be separated and applied to LLMs as well, benefiting both.

By explicitly classifying whether the context is useful for answering the query, the confidence of context measured by RE provides various decoding strategies. If the retrieved context set is irrelevant, we can choose to classify the query as "unanswerable", while maintaining most of the accuracy for the answerable set. Additionally, if a low-confidence context set is retrieved, which will likely result in wrong answers by parroting the context as is (Jia and Liang, 2017), we can instead selectively leverage the LLM's parametric knowledge to improve answer accuracy in most cases.

The main contributions of our work are:

1. We propose a new framework called **RE-RAG** by adding an external Relevance Estimator (RE) module. We further suggest a weak supervision training method that can train RE without explicit labeled data on question-context compatibility. (§2.2)

2. We demonstrate that RE-RAG, enhanced with RE, significantly improves upon the existing RAG. Addtionally, we show that RE trained on a small language model can improve the answer performance of LLMs. (§4.1)

3. We propose to use the confidence level of the context set measured by RE to answer "unanswerable" for unanswerable context sets with minimal negative effects, or to complement LLM's parametric knowledge. (§5.1)

2 Method

In this section, after reviewing the traditional RAG framework, we present the RE-RAG model combined with our relevance estimator.

2.1 Traditional RAG overview

Retriever Retriever searches for information in an external knowledge base and returns a related context set $\mathbf{C}_i$. In general, RAG systems use a bi-encoder type retriever such as DPR (Karpukhin et al., 2020), which is effective and fast in retrieving information. A question $\mathbf{q}_i \in \mathbf{Q}$ and a context $\mathbf{c}_j \in \mathbf{C}_i$ are input to the encoder independently to obtain an embedding of $\mathbf{Emb}_q = Encoder(q_i)$, $\mathbf{Emb}_c = Encoder(c_j)$. The similarity score $\mathbf{S}_{i,j} = \mathbf{Emb}_q \cdot \mathbf{Emb}_c$ is calculated from the obtained embedding and then used to perform top-k context retrieval.

Generator Generators that utilize the sequence-to-sequence model typically take a question and context as input and produce an answer $\mathbf{y}_{i,j}$ with probability $\mathbf{P}_G(\mathbf{y}_{i,j}|\mathbf{q}_i, \mathbf{c}_j)$.

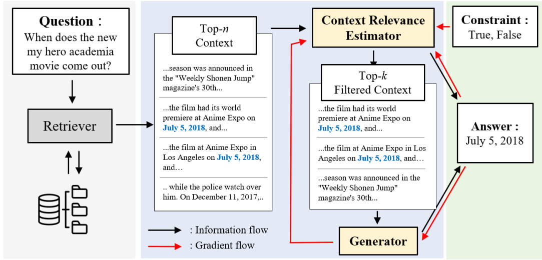

Figure 1: Overview of our proposed RE-RAG framework. The black lines represent the flow of information and the red lines represent the flow of gradients.

Answer marginalization RAG (Lewis et al., 2020b) introduced the answer generation models of RAG-sequence and RAG-token. We focus on the RAG-sequence model which marginalizes probability of $y_l \in \mathcal{Y}_i$ where $\mathcal{Y}_i$ is an aggregated set of $\mathbf{y}_{i,j}$. which achieves higher performance than the RAG-token model and ensures the interpretability of the answer generation process. Individually generated answers $\mathbf{y}_{i,j}$ per $\mathbf{c}_j$ are marginalized as $\mathbf{y}_l$ using the similarity score $\mathbf{S}_{i,j}$ as shown in eq.(2).

$$\mathbf{P}_R(\mathbf{S}_{i,j}) = \frac{e^{\mathbf{S}_{i,j}}}{\sum_k e^{\mathbf{S}_{i,k}}} \quad (1)$$

$$\mathbf{P}_a(\mathbf{y}_l|\mathbf{q}_i, \mathbf{C}_i) = \sum_j \mathbf{P}_R(\mathbf{S}_{i,j}) \cdot \mathbf{P}_G(\mathbf{y}_l|\mathbf{q}_i, \mathbf{c}_j) \quad (2)$$

2.2 RE-RAG framework

The retriever similarity score $\mathbf{S}_{i,j}$ is trained to achieve high recall when retrieving multiple contexts, however, it was not initially designed to provide fine-grained relevancy score $\mathbf{P}_R(\mathbf{S}_{i,j})$ for aiding RAG generation steps in eq.(2). To address this issue, we propose a relevance estimator (RE) that re-ranks contexts and provides precise relevance scores to the generator.

Relevance Estimator Relevance estimator (RE) measures the relevance between a question and context. We utilize a similar architecture to Nogueira et al. (2020) which utilizes a sequence-to-sequence model as a passage reranker.

Our RE receives the same input of question and context as the generator, but is trained to generate a **classification token** ("true" or "false") based on the relevance of the context to the input question. We normalize the probability of generating "true" and "false" tokens to get the final probability of generating the classification token. The obtained probability of a "true" token can independently be an indicator of the relevance of a single context to a given question. When comparing between multiple contexts, the "true" token probability can be converted to logit and used as the relevance score of the retrieved context.

$$\mathbf{RE}_{i,j} = \frac{\mathbf{P}(\text{“}true\text{”}|\mathbf{q}_i, \mathbf{c}_j)}{\mathbf{P}(\text{“}true\text{”}|\mathbf{q}_i, \mathbf{c}_j) + \mathbf{P}(\text{“}false\text{”}|\mathbf{q}_i, \mathbf{c}_j)} \quad (3)$$

Reranking of contexts by relevance With the trained relevance estimator RE, we can rerank contexts in the initial retrieved set $\mathbf{C}_i$ by their relevance and only take top-k contexts to redefine $\mathbf{C}_i$ before the answer-generation step. With a precise relevance score from RE, we can expect the RE-RAG to be more efficient, i.e. stronger performance with lower computation (see §4.2).

Answer marginalization with context RE The question and context are concatenated and input to the generator model, and the generator generates $\mathbf{P}_G(\mathbf{y}_{i,j}|\mathbf{q}_i, \mathbf{c}_j)$ per question. We replace the probability distribution $\mathbf{P}_R(\mathbf{S}_{i,j})$ in eq.(2) with the relevance scores from context RE to form eq.(6) as following:

$$\sigma(\mathbf{RE}_{i,j}) = \log\left(\frac{\mathbf{RE}_{i,j}}{1 - \mathbf{RE}_{i,j}}\right) \quad (4)$$

$$\mathbf{P}_{\mathbf{RE}}(\mathbf{q}_i, \mathbf{c}_j) = \frac{e^{\sigma(\mathbf{RE}_{i,j})}}{\sum_k e^{\sigma(\mathbf{RE}_{i,k})}} \quad (5)$$

$$\mathbf{P}_a(\mathbf{Y}_i|\mathbf{q}_i, \mathbf{C}_i) = \sum_j \mathbf{P}_{\mathbf{RE}}(\mathbf{q}_i, \mathbf{c}_j) \cdot \mathbf{P}_G(\mathbf{y}_{i,j}|\mathbf{q}_i, \mathbf{c}_j). \quad (6)$$

We can expect higher performance with the marginalized answer $\mathbf{y}_l$ if RE can provide an accurate relevance distribution $\mathbf{P}_{\mathbf{RE}}$ (see §5.2).

2.3 Joint training of RE-RAG

We propose to utilize three different types of losses to train RE-RAG with our proposed relevance estimator. First, to train the generator model, we use a loss that combines the commonly used negative likelihood loss for ground truth $\mathbf{a}_i$ with a probability that represents the relevance of the question and context.

$$\mathbf{L}_{\text{gen}} = -\sum_{i,j} \log\left(\mathbf{P}_{\mathbf{RE}}(\mathbf{q}_i, \mathbf{c}_j) \cdot \mathbf{P}_G(\mathbf{a}_i|\mathbf{q}_i, \mathbf{c}_j)\right) \quad (7)$$

$\mathbf{L}_{\text{gen}}$ simultaneously adjusts the probability of generating the classification token for the relevance estimator while training the generator.

Second, to obtain a learning signal for training the relevance estimator, we calculate the log-likelihood loss of the generator per retrieved context and compute its distribution across contexts as follows:

$$\mathbf{F}_{i,j} = \log(\mathbf{P}_G(\mathbf{a}_i|\mathbf{q}_i, \mathbf{c}_j)) \quad (8)$$

$$\mathbf{Q}_G(\mathbf{q}_i, \mathbf{c}_j) = \frac{e^{\mathbf{F}_{i,j}}}{\sum_k e^{\mathbf{F}_{i,k}}}. \quad (9)$$

The log-likelihood loss varies depending on whether an answer can be inferred from the input context. Therefore, applying the softmax function to the log-likelihood loss values yields a probability distribution that represents the relevance between the given set of contexts and the question. We do not leverage any labeled data that entails the relevance of questions and contexts.

$\mathbf{Q}_G(\mathbf{q}_i, \mathbf{c}_j)$ represents relative relevance between $\mathbf{q}_i$ and $\mathbf{c}_j$

We calculate the KL-divergence loss between the probability distributions of the generator and the RE, and use this loss to train the model.

$$\mathbf{L}_{\text{re}} = D_{\text{KL}}(\mathbf{P}_{RE}(\mathbf{q}_i, \mathbf{c}_j)||\mathbf{Q}_G(\mathbf{q}_i, \mathbf{c}_j)) \quad (10)$$

Lastly, in addition to applying a training loss on the probability of generating the classification token, we need to set an additional loss to prevent the RE from generating tokens other than the classification token. To do this, we utilize the additional loss as the sum of the probability of RE of generating all tokens other than classification token.

$$\mathbf{L}_{\text{tok}} = \sum_{t \in T \setminus \{\text{"true","false"}\}} \mathbf{P}(t|\mathbf{q}_i, \mathbf{c}_k) \quad (11)$$

To train an effective system, the two models are trained jointly utilizing all three losses as follows:

$$\mathbf{L}_{\text{tot}} = \mathbf{L}_{\text{gen}} + \alpha_1 \mathbf{L}_{\text{re}} + \alpha_2 \mathbf{L}_{\text{tok}} \quad (12)$$

where α_1 and α_2 are hyperparameters that act as scaling factors to balance the impact of each loss.

3 Experimental Setup

We evaluated the performance of our model on an open-domain QA dataset. In this section, we describe the dataset we used in our experiments and the details of our experiments.

Model	Extra	Generator	NQ	TQA	# Contexts
Small language models ($\leq 2B$)					
RAG (Lewis et al., 2020b)	-	445M	44.5	56.8	50
FiD$_{base}$ (Izacard and Grave, 2021b)	-	220M	48.2	65.0	100
FiD$_{large}$ (Izacard and Grave, 2021b)	-	770M	51.4	67.6	100
FiD-KD$_{base}$ (Izacard and Grave, 2021a)	-	220M	50.1	<u>69.3</u>	100
FiD-KD$_{large}$ (Izacard and Grave, 2021a)	-	770M	<u>54.4</u>	**72.5**	100
ReAtt (Jiang et al., 2022)	-	770M	**54.7**	-	100
FiD-KD$_{base}$ (Izacard and Grave, 2021a)	-	220M	48.6	67.4	25
FiD-KD$_{large}$ (Izacard and Grave, 2021a)	-	770M	53.9	**71.2**	25
R2-D2 (Fajcik et al., 2021)	125M	1.04B	**55.9**	69.9	25
RE-RAG$_{base}$	220M	220M	49.9	68.2	25
RE-RAG$_{mixed}$	770M	220M	51.4	69.5	25
RE-RAG$_{large}$	770M	770M	<u>54.0</u>	<u>70.2</u>	25
Large language models ($\geq 7B$)					
Self-RAG$_{7B}$ (Asai et al., 2023)	-	7B	-	66.4	5
Self-RAG$_{13B}$ (Asai et al., 2023)	-	13B	-	<u>69.3</u>	5
Llama2$_{7b}$ + RE	770M	7B	<u>45.7</u>	67.1	5
Llama2$_{13b}$ + RE	770M	13B	**46.6**	**70.8**	5
RA-DIT (Lin et al., 2023)	-	65B	43.9	75.1	10
Llama2$_{70b}$ + FiD-KD$_{ret}$	-	70B	38.1(40.7)	63.5(66.3)	10
Llama3$_{70b}$ + FiD-KD$_{ret}$	-	70B	39.5(46.8)	68.1(72.1)	10
ChatGPT + FiD-KD$_{ret}$	-	175B	42.9(45.9)	69.0(70.7)	10
Codex + REPLUG LSR (Shi et al., 2023)	-	175B	45.5	**77.3**	10
Llama3$_{8b}$ + RE	770M	8B	<u>49.6</u>	73.0	10
Llama2$_{70b}$ + RE	770M	70B	48.0	72.4	10
Llama3$_{70b}$ + RE	770M	70B	**50.8**	<u>75.5</u>	10
ChatGPT + RE	770M	175B	49.3	72.6	10

Table 1: EM scores on Natural Questions and TriviaQA datasets. The parameters of the generator and the extra module that evaluates a given context are listed separately. # Contexts refer to the number of contexts utilized for inference. For an effective comparison, we divided the groups based on the size of the generator model and the number of contexts utilized for inference. Our experiment results on all LLMs ($\geq 7B$) follow traditional RAG method, which aggregates answers by context. In the case of applying the FiD-KD retriever to LLMs, the numbers in the (right) represent the RALM method, which concatenates contexts to generate answers. We provide this extra result to fairly compare with FiD-KD retriever as it did not provide a suitable relevance score for the traditional RAG method to perform well. The bold is the best score in each group, and the underline is the second best.

3.1 Dataset

We evaluate our performance on two open-domain QA datasets:Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017). To train and evaluate our model, we utilize the context datasets retrieved for each question from NQ and TQA, as used in FiD-KD (Izacard and Grave, 2021a) and Akari (Asai et al., 2022). The dataset includes the top-20 training contexts, while the dev and test sets contain the top-100 contexts retrieved by the retriever. We used 20 contexts for training and the top-25 contexts extracted by the RE from the top-100 retrieved contexts for inference.

Natural Questions Natural Questions (Kwiatkowski et al., 2019) is a dataset of real questions asked by users on the web. The dataset consists of questions collected from the web, a long answer that can be viewed as gold context for the question, and a short answer with a short span. The open-domain QA version dataset of Natural Questions is a dataset that collects only questions where the answer span of the short answer is 5 tokens or less in length. We use the NQ-open dataset.

TriviaQA TriviaQA (Joshi et al., 2017) is a dataset of question-answer pairs collected from trivia enthusiasts. Each question and answer in the dataset has been reviewed by human annotators. We want to use the unfiltered version of TriviaQA dataset.

3.2 Evaluation Metric

The predicted answers are evaluated using **EM score**, a commonly used metric as in Izacard and Grave (2021b), Rajpurkar et al. (2016). The generated answers are normalized (e.g., lowercase, punctuation, article stripping) and compared to the correct answers in the dataset. We consider a generated answer to be correct if it exactly matches one of the correct answers in the given dataset after normalization.

3.3 Baseline

We investigate whether the performance of RE-RAG is competitive with that of the FiD (Izacard and Grave, 2021b)-based system. FiD has achieved excellent performance on the Question-Answering task, and the FiD-based application system also outperforms the RAG (Lewis et al., 2020b)-based system on the QA task.

We consider an additional baseline to compare the performance of RE when applied to LLMs. We compare the performance of RE and FiD-KD retriever when applied to LLMs. When applying the FiD-KD retriever to LLMs, we compared two methods: traditional RAG, which uses the retriever similarity score to perform answer marginalization, and RALM, which concatenates all context. Furthermore, we compare our performance with other studies (Asai et al., 2023; Lin et al., 2023; Shi et al., 2023) that have implemented RAG in LLMs.

3.4 Model

The two components of our framework, RE and the generator, utilize the T5 model (Raffel et al., 2020). We utilize the T5-base, T5-large models, and explore three different model sizes depending on the combination of the two models.

Additionally, we utilize Llama2 (7B, 13B, 70B), Llama3 [1] (8B, 70B), and ChatGPT ("gpt-3.5-turbo-0125" version) as generators to assess if RE brings performance improvements when applied to LLMs. In our experiments, the LLMs used as generators are not fine-tuned for the downstream task.

4 Experiment Results

We investigate the QA performance of the RAG system with our newly proposed relevance estimator (RE). In addition to the QA performance of the whole system, we also examine the performance of the RE independently.

4.1 Main Results

The overall accuracy of our system on the two datasets (NQ and TQA) is shown in Table 1. Compared to the traditional RAG, our system, RE-RAG, performs better despite having the same total number of parameters. Our proposed RE improves the reliability of the RAG system by more accurately measuring the relevance between question and context. Our model performed competitively with models based on FiD structures(Izacard and Grave, 2021a; Jiang et al., 2022; Fajcik et al., 2021).

The accuracy of the RE module when applied to Large Language Models (LLMs) is shown at the bottom of Table 1. We only included the RAG-based model in our comparison because the FiD-based model is not applicable to LLMs due to structural differences. The RE module outperforms the FiD-KD retriever when applied to LLMs. When the RE module is applied to Llama2, it surpasses the Self-RAG, where the LMs themselves inspect the retrieved context and generated answers. In TQA, REPLUG with Codex scores slightly higher. The performance of TQA seems to depend more on the generator model than NQ (see Figure 2 for a related discussion), and we believe that this is the reason for the performance difference with Codex. Our model performs better on NQ, which is a more knowledge intensive task.

Dataset	Model	Recall@k			
		R@1	R@5	R@10	R@20
NQ	FiD-KD	49.4	73.8	79.6	84.3
	MonoT5$_{large}$	46.2	72.4	80.1	84.7
	RE-RAG$_{base}$	59.5	77.8	82.7	85.5
	RE-RAG$_{large}$	**61.9**	**79.4**	**83.6**	**86.4**
TQA	FiD-KD	60.1	77.0	80.9	83.6
	MonoT5$_{large}$	64.7	79.7	82.9	84.8
	RE-RAG$_{base}$	67.0	81.5	83.6	85.4
	RE-RAG$_{large}$	**70.4**	**82.2**	**84.4**	**86.1**

Table 2: Performance of RE as a re-ranker. The re-ranking performance for the top-100 contexts retrieved by the FiD-KD retriever is denoted by recall@k.

Dataset	Model	Recall	Precision	F1
NQ	FiD-KD	73.2	21.9	33.7
	MonoT5$_{large}$	10.3	31.0	15.5
	RE-RAG$_{base}$	51.3	33.9	40.9
	RE-RAG$_{large}$	45.9	38.3	41.7
TQA	FiD-KD	64.3	24.5	35.5
	MonoT5$_{large}$	27.2	34.2	30.3
	RE-RAG$_{base}$	38.9	46.7	42.5
	RE-RAG$_{large}$	39.0	43.2	41.0

Table 3: Classification results for context sets that do not contain an answer within the top-25 context set. We used cosine similarity for FiD-KD's retriever and "true" token probability for our method and MonoT5.

4.2 Performance of RE as a reranker and unanswerable set classifier

Table 2 shows the performance of our proposed RE-RAG's RE as a reranker. For the Recall@k metric, we use the retrieval accuracy used by DPR (Karpukhin et al., 2020), FiD-KD (Izacard and

[1] https://github.com/meta-llama/llama3

Dataset	Model	Score	Answerable context set O	X
NQ	RE-RAG$_{base}$	FiD-KD	58.3 → 32.7	73.4
	RE-RAG$_{base}$	RE	58.3 → 54.9	51.3
	RE-RAG$_{large}$	FiD-KD	61.5 → 34.9	71.3
	RE-RAG$_{large}$	RE	61.5 → 57.9	45.9
TQA	RE-RAG$_{base}$	FiD-KD	78.7 → 51.2	63.5
	RE-RAG$_{base}$	RE	78.7 → 77.0	38.9
	RE-RAG$_{large}$	FiD-KD	80.4 → 51.6	62.7
	RE-RAG$_{large}$	RE	80.4 → 77.9	39.0

Table 4: We examine whether RE can successfully identify unanswerable scenarios where retrieved contexts do not hold true answers. **O** refers to the retrieval context set that contains true answers and **X** refers to the set without which we dim as *unanswerable*. Under the **X**, we denote the classification accuracy for the unanswerable set. Under the **O**, we denote the accuracy change as the RE thresholding will inevitably classify the context sets with answers as unanswerable. Left of the arrow denotes original accuracy on **O** and the right denotes accuracy after RE score thresholding.

Grave, 2021a), and ColbertQA (Khattab et al., 2021). Although the comparison retriever has been enhanced through knowledge distillation methods using FiD attention scores, our proposed RE still demonstrated superior performance. In particular, RE performs better as the number of contexts decreases, which means that RE is more efficient when there are fewer contexts to utilize.

Table 3 shows the performance of the context relevance estimator (RE) as a "unanswerable" set classifier. "unanswerable" set means that the context set of the top-25 contexts does not contain a gold answer in any context. For classification, we used the cosine similarity score of the hidden representation of the question and context for retriever and the probability of generating a "true" token by the model for RE and MonoT5 (Nogueira et al., 2020). For the optimal threshold, we searched for the value that maximizes F1 score in steps of 0.1 from 0.5 to 0.9 at dev set.

Our RE showed better "unanswerable" set classification performance than FiD-KD retriever or MonoT5 based on F1 score. Looking at the detailed performance, we found that the retriever performed better for recall, but the RE performed better for precision. This is because the retriever classified a large number of context sets as all "unanswerable" sets, while our proposed RE showed a good balance between precision and recall.

5 Analysis

5.1 Exploring decoding strategies in low confidence context sets

In this section, we review two strategies that can be used when a context set with a low confidence score is retrieved. The confidence score for a context set is determined using the maximum value of the "true" token probability computed by RE for the contexts within the set. We examine the strategy of answering "unanswerable" when a low confidence context set is returned in a small Language Model (sLM), where parametric knowledge is scarce. Additionally, we examine the strategy of directly utilizing parametric knowledge in Large Language Models (LLMs), where parametric knowledge is abundant.

Classify as "unanswerable" Table 4 shows the change in accuracy after letting the model respond with "unanswerable" when the retrieved context set has low confidence. For the confidence threshold value that determines whether the model should respond with "unanswerable", we chose the value that optimizes the classification performance as determined in Table 3. We evaluate the accuracy by dividing the entire test set into answerable sets, which contain at least one gold answer in the context set, and unanswerable sets, which contain none.

Our RE model shows relatively minor accuracy loss on the answerable set when responding with "unanswerable" for context sets measured with low confidence, but gains significant ability on the unanswerable set. In contrast, the FiD-KD retriever loses a substantial amount of accuracy on the answerable set when it responds with "unanswerable" for low-confidence context sets, resulting in a larger negative effect compared to our model. If we want to preserve the answerable set accuracy of the FiD-KD retriever, its ability to classify "unanswerable" is significantly reduced compared to RE (see Appendix E).

Selectively using parametric knowledge We further explore how we can effectively utilize the rich parametric knowledge of LLMs. When the confidence of the retrieved context is low, we examine a mixed strategy that optionally bypasses the context and relies solely on the parametric knowledge of the largest model to generate the correct answer. For the confidence threshold value that determines whether the model should answer using only parametric knowledge, we selected the

P-Generator	R-Generator	NQ	TQA
Llama2_{70b} (NQ: 31.1/TQA: 64.3)	Llama2_{7b}	46.2 → 45.9(-0.3)	68.0 → 69.3(+1.3)
	Llama2_{13b}	47.3 → 46.5(-0.8)	71.5 → 72.1(+0.6)
	Llama2_{70b}	48.0 → 46.9(-1.1)	72.4 → 72.9(+0.5)
Llama3_{70b} (NQ: 41.3/TQA: 75.1)	Llama3_{8b}	49.6 → 49.8(+0.2)	73.0 → 75.4(+2.4)
	Llama3_{70b}	50.8 → 50.8(-)	75.5 → 76.7(+1.2)
ChatGPT (NQ: 37.7/TQA: 72.0)	ChatGPT	49.3 → 49.3(-)	72.6 → 73.6(+1.0)

Table 5: Change in EM scores when utilizing the LLM's parametric knowledge for low-confidence context sets. P-Generator model, which relies solely on its parametric knowledge, has EM scores shown below its name. R-Generator refers to a model that utilizes RAG. For both datasets, the confidence score threshold for model selection is set to 0.7. See appendix D for results on FiD-KD retriever.

Model	NQ	TQA
Baseline	39.5	54.9
Baseline w/ RE score	43.1	60.1
Baseline w/ RE rerank	46.8	63.9
Baseline w/ RE rerank, score	49.6	67.8
RE-RAG_{base}	49.9	68.2

Table 6: An ablation study to decompose the effect of RE in RE-RAG. We compared the traditional RAG model without RE, with reranking of RE (RE rerank), with RE score in answer generation (RE score), and with both (RE rerank, score).

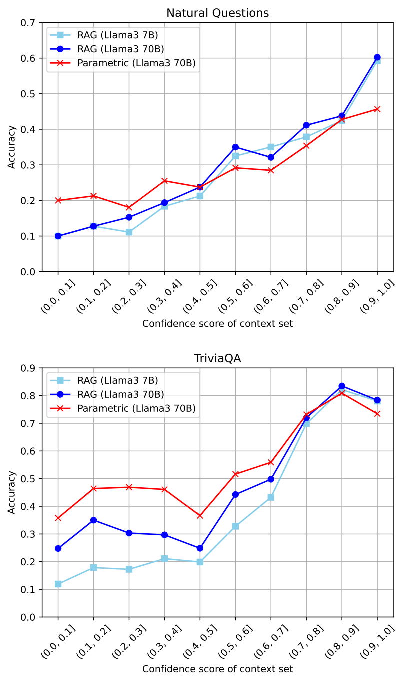

Figure 2: The relationship between confidence score and accuracy by model size. RAG means that the model utilizes contextual knowledge and Parametric means that the model utilizes only parametric knowledge without external knowledge.

value that optimizes classification performance as determined in Table 3. For each type of model, we utilize the one with the largest number of parameters as the parametric knowledge base.

Table 5 shows the change in accuracy when decoding the answer using the mixed strategy. In most cases, our strategy achieves accuracy gains in TQA without significant losses in NQ, except in cases where parametric knowledge is particularly scarce, such as in NQ on Llama2. NQ is a more knowledge-intensive task compared to TQA, where there is less benefit from utilizing parametric knowledge.

When parametric knowledge can be used effectively, the mixed strategy achieves larger gains in smaller models, and the performance gap narrows compared to larger models. Figure 2 illustrates the relationship between confidence score and accuracy by model size. At high confidence scores on the TQA dataset, small size models achieve similar accuracy to large size models. At low confidence scores, the difference in performance between small and large models becomes more pronounced. When using small size models, higher efficiency can be achieved by utilizing retrieval augmented generation only when a high confidence context set is retrieved, and selectively leveraging the parametric knowledge of large size models when a low confidence context set is retrieved.

5.2 Ablation Study

Effectiveness of RE We perform an ablation study to investigate the effectiveness of the added RE in `RE-RAG`. The effect of our proposed RE is twofold. First, it performs better re-ranking than the retriever, selecting more accurate context and passing it to the generator. Second, it calculates a more accurate relevance score than retriever's similarity score and uses it in the answer marginalization process. In Table 6, the performance of methods with each component of the RE added is presented, using a model that was trained with only the T5-base generator, after removing the RE, as the baseline.

We construct the following experiment to isolate the two effects. First, we apply the top 25 contexts from retriever and their similarity scores to the baseline model. Next, there are the top-25 contexts from the retriever with the RE's score applied (RE score) and the top-25 contexts from the RE with the retriever's similarity score applied (RE rerank). Finally, we compare the performance of applying

Model	NQ	TQA
Baseline	0.435	0.561
- normalization	0.0005	0.0002

Table 7: Average value of the probability that RE generates the "true" token for answerable contexts when the normalization process is removed.

Model	NQ	TQA
Baseline	49.1	67.8
- L_{re}	48.0	66.7

Table 8: Difference in EM scores on the dev set when L_{re} is removed from the training process.

the RE's top-25 contexts and score to the baseline model (RE rerank, score).

Both effects of the RE are found to be significant in improving the performance of the baseline model. This shows that not only the quality of the context input to the generator plays an important role, but also the score, which means the importance of each context.

Remove training components We investigate the impact of removing the regularization process in eq.(3) on the classification performance of RE while training on the RE-RAG$_{base}$ model. Table 7 shows how the "true" token probability level output by the RE changes when the normalization process is removed. It can be seen that when the normalization process is removed, RE can only perform the function of re-ranking but loses the function of measuring confidence. This is because the normalization process allows the model to adjust its output strictly between "true" and "false" tokens.

Table 8 shows the difference in EM scores on the dev set when L_{re} is removed from the training process. We observed that removing L_{re} from the training process decreases answer performance. We believe that L_{re} contributes to achieving more optimal performance by using loss information from generator to directly propagate the relative importance of contexts to the RE.

6 Related Works

Previous research has shown that the performance of Question Answering systems can be improved by utilizing external knowledge about questions (Chen et al., 2017). Methods for more accurate retrieval of external knowledge (Karpukhin et al., 2020; Khattab et al., 2021; Gao and Callan, 2022) have been studied to make these systems more efficient. In open-domain QA, models that extract and use answers from retrieved documents have been studied (Karpukhin et al., 2020; Khattab et al., 2021; Cheng et al., 2021), but studies that utilize generative models such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020a) have become more common (Lewis et al., 2020b; Izacard and Grave, 2021b). RAG and FiD achieved powerful performance in open-domain QA using different methods. Subsequently, models (Izacard and Grave, 2021a; Fajcik et al., 2021; Singh et al., 2021; Jiang et al., 2022) that leverage and improve upon the structural advantages of FiD have been proposed. For Atlas (Izacard et al., 2022), state-of-the-art performance was achieved through an improved retriever (Izacard et al., 2021) and scaling up the model. In the case of RAG, there is a study that improved performance by introducing a BERT (Devlin et al., 2019)-based reranker (Glass et al., 2022), but it utilized additional data and high-quality label data when training the reranker.

Recently, large language models (LLMs) such as GPT (Brown et al., 2020) and Llama (Touvron et al., 2023), which have been developed in recent years, face limitations with FiD methods that require encoded data. Consequently, research on RAG models, which can directly input context, has received renewed attention. (Asai et al., 2023; Lin et al., 2023; Shi et al., 2023) These approaches have achieved performance improvements by training a retriever, which can also be applied to LLMs, or by performing the review of questions and context within the model itself.

7 Conclusion

We propose the RE-RAG framework, which extends traditional RAG by incorporating RE that can measure the relative relevance and confidence of contexts. We demonstrate that the RE-RAG framework can enhance the performance of traditional RAG. We show that the RE module, as a detachable component, can be combined with modern large language models (LLMs) to improve their performance. Furthermore, we exploree some decoding strategies that leverage the confidence information measured by the RE module to either answer "unanswerable" or selectively utilize the parametric knowledge of the LLMs when a low confidence context set is retrieved. We hope that our research will inspire the exploration of various additional modules for retrieval-augmented generation.

8 Limitation

Our research has primarily focused on improving answer performance in single-hop QA tasks. We have not sufficiently verified the effectiveness of our proposed framework in multi-hop QA tasks. We believe that in the future, we can explore whether the RE-RAG framework can be extended to multi-hop QA.

In our work, we explored a decoding strategy that measures with confidence whether a context is truly useful for a query and classifies low confidence contexts as unanswerable. However, a truly unanswerable query is one where the query cannot be adequately answered even when utilizing the model's parametric knowledge. We believe that future research needs to be conducted to detect whether the parametric knowledge has knowledge that can adequately answer the query in order to finally classify the unanswerable problem.

References

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive nlp tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2226–2243.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

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.

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.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. UnitedQA: A hybrid approach for open domain 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 3080–3090, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-d2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 854–870.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard and Edouard Grave. 2021a. Distilling knowledge from reader to retriever for question answering. In *ICLR 2021-9th International Conference on Learning Representations*.

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

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*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Zhengbao Jiang, Luyu Gao, Zhiruo Wang, Jun Araki, Haibo Ding, Jamie Callan, and Graham Neubig. 2022. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer.

In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2336–2349.

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.

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.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding. *arXiv preprint arXiv:2305.04757*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

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*.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981.

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*.

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

A Dataset Statistics

Table 8 shows the statistics for the Natural Questions and TriviaQA unfilitated datasets we used.

Dataset	Train	Dev	Test
Natural Questions	79,168	8,757	3,610
TriviaQA	78,785	8,837	11,313

Table 9: Dataset statistics for Natural Questions and TriviaQA

Question	Context	Gold Answer	"True" prob
who played mark on the show the rifleman	...**Mark McCain is the son of fictitious rancher Lucas McCain in the ABC Western television series "The Rifleman,"** starring Chuck Connors, which ran from 1958 to 1963. Singer/actor and former Mouseketeer **Johnny Crawford was cast in the role** and...	John Ernest Crawford	0.987
when does the cannes film festival take place	...2017 **Cannes Film Festival The 70th Cannes Film Festival took place from 17 to 28 May 2017**, in Cannes, France ...	Cannes, France, usually in May	0.994
how many strong verbs are there in german	...**Germanic strong verbs are commonly divided into 7 classes**, based on the type of vowel alternation. This is in turn based mostly...	more than 200, more than 200 strong	0.949
how many episodes of corrie has there been	...The show airs six times a week: Monday, Wednesday and Friday 7:30-8 pm and 8:30-9 pm. Since 2017, **ten sequential classic episodes** of the series from 1986...	9,436	0.147

Table 10: The relevance measure of the question and context output by the RE. The first two show relevant contexts that contain the correct answer even if the context does not include exactly the same surface form compared to the true answer. The last two examples show irrelevant contexts that actually have high overlap with question tokens, however, without pertaining the correct answer.

B Training Details

We used T5-base with a parameter size of 223M and T5-large model with a parameter size of 770M as modulators in all experiments. We trained the RE-RAG$_{base}$ system on 4 A6000 GPUs, while RE-RAG$_{mixed}$ and RE-RAG$_{large}$ were trained on 2 A100 and 4 A100 GPUs, respectively.

We used a constant learning rate of $10^{-}4$ for all sizes of RE-RAG systems. We used AdamW as the optimizer and weight decay was $10^{-}3$. For batch size, we used gradient accumulation for all sizes of models, resulting in an effective batch size of 64. For the hyperparameters that balance the proposed losses, we utilized the default value of 1 for both α_1 and α_2. We did not explore hyperparameters that achieve better performance due to time and limited computing resources.

For model selection, we evaluated every 1 epoch and selected the case with the highest answer accuracy of the dev set. The dev set answer accuracy was measured using the top-10 context of the RE. Since the answer accuracy of the top-10 context of the RE is similar to the answer accuracy of the top-25 context, this helped to save computational resources and time while still producing valid results.

C Effectiveness of the RE

We perform a qualitative analysis to see if our proposed relevance estimator (RE) is effectively classifying relevant contexts. Table 3 shows a few contexts in the NQ test set.

Some of the contexts that the RE predicts are highly relevant to the question even when they do not contain the exact ground truth answer. The first few examples in Table 3 are examples that are categorized as true context because they contain phrases that are semantically equivalent to the correct answer albeit not having the exact same form in the context. This shows that although the RE is trained to measure the relevance of a question to a context through a limited set of ground truth answers, it is actually capable of measuring a broader range of relevance.

P-Generator	R-Generator	NQ	TQA
Llama2$_{70b}$ (N31.1/T64.3)	Llama2$_{7b}$	36.1 → 35.8(-0.3)	58.4 → 62.8(+4.4)
	Llama2$_{13b}$	38.8 → 36.9(-1.9)	64.9 → 65.4(+0.5)
	Llama2$_{70b}$	40.7 → 37.4(-3.3)	66.3 → 66.2(-0.1)
Llama3$_{70b}$ (N41.3/T75.1)	Llama3$_{8b}$	38.2 → 42.1(+3.9)	57.6 → 66.9(+9.3)
	Llama3$_{70b}$	46.8 → 45.6(-1.2)	72.1 → 74.0(+1.9)
ChatGPT (N37.7/T72.0)	ChatGPT	45.9 → 43.2(-2.7)	70.7 → 72.1(+1.4)

Table 11: The change in EM score when using the cosine similarity score of the FiD-KD retriever for the confidence score, when utilizing LLM's parameter knowledge for a set of low confidence contexts. The thresholds were set to 0.7 for NQ and 0.6 for TQA, as specified in Table 3.

In addition to the examples above, there are cases where the RE misclassified contexts as containing the correct answer. As shown in the example in Table 10, the RE classified the context containing "the number of classes of strong verbs in German" as the correct context for the question about "the number of strong verbs in German", which means that our RE is still limited in its ability to capture the fine-grained meaning of the question in the retrieved context. On the other hand, in the last example, for the question about "the number of

Dataset	Type	Threshold				
		0.5	0.6	0.7	0.8	0.9
NQ	Answerable	61.3	56.2	**34.9**	6.4	0.0
	Unanswerable	2.3	27.8	**71.3**	97.2	99.8
TQA	Answerable	77.3	**51.6**	9.2	0.1	0.0
	Unanswerable	14.3	**62.7**	94.7	100.0	100.0

Table 12: Performance variation of FiD-KD retriever on answerable and unanswerable sets for different thresholds.

episodes", it succeeded in classifying the context containing "the number of classical episodes" as an incorrect context.

D Selectively using parametric knowledge with FiD-KD

Table 11 shows the change in EM score when applying the mixed decoding strategy, using the cosine similarity score of the FiD-KD retriever as the confidence score. For small parameter generators, the EM score is low when applying the FiD-KD retriever to LLMs, which results in a high gain when utilizing parametric knowledge of large parameter models. However, since the classification performance of the FiD-KD retriever is lower than that of RE, even utilizing parametric knowledge does not significantly outperform the baseline performance of parametric knowledge. Especially for more knowledge-intensive tasks such as NQ, the performance loss is substantial.

E FiD-KD retriever's performance in "unanswerable" scenarios

Table 12 shows the performance of the FiD-KD retriever in unanswerable scenarios according to different threshold values. For the FiD-KD retriever, it is observed that while trying to maintain performance on the answerable set, the classification ability on the unanswerable set significantly decreases.