Prompt-Guided Retrieval Augmentation for Non-Knowledge-Intensive Tasks

Zhicheng Guo[1, 3]*, **Sijie Cheng**[1, 2, 3, 5], **Yile Wang**[2], **Peng Li**[2, 4†], and **Yang Liu**[1,2,3,4†]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China
[3]Beijing National Research Center for Information Science and Technology, Beijing, China
[4]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[5]School of Computer Science, Fudan University, Shanghai, China

Abstract

Retrieval-augmented methods have received increasing attention to support downstream tasks by leveraging useful information from external resources. Recent studies mainly focus on exploring retrieval to solve knowledge-intensive (KI) tasks. However, the potential of retrieval for most non-knowledge-intensive (NKI) tasks remains under-explored. There are two main challenges to leveraging retrieval-augmented methods for NKI tasks: 1) the demand for diverse relevance score functions and 2) the dilemma between training cost and task performance. To address these challenges, we propose a two-stage framework for NKI tasks, named PGRA. In the first stage, we adopt a *task-agnostic retriever* to build a shared static index and select candidate evidence efficiently. In the second stage, we design a *prompt-guided reranker* to rerank the nearest evidence according to task-specific relevance for the reader. Experimental results show that PGRA outperforms other state-of-the-art retrieval-augmented methods. Our analyses further investigate the influence factors to model performance and demonstrate the generality of PGRA. Codes are available at https://github.com/THUNLP-MT/PGRA.

1 Introduction

Retrieval-augmented methods aim at enhancing dense models with non-parametric indices to better leverage external knowledge (Borgeaud et al., 2022; Izacard et al., 2022; Wang et al., 2022). By decoupling knowledge storage from model parameters, retrieval-augmented methods can achieve comparable or better performance than large-scale pre-trained models with orders of magnitude less parameters on tasks such as language modeling (Khandelwal et al., 2020; Guu et al., 2020; Borgeaud et al., 2022) and question answering (Lee et al., 2019; Karpukhin et al., 2020; Izacard and Grave, 2021). Moreover, as external knowledge is stored in the non-parametric index, knowledge can be updated simply by replacing the index without further training (Izacard et al., 2022). Therefore, retrieval-augmented methods have attracted increasing interest in recent years and achieved promising results in various natural language processing tasks (Zhang et al., 2018; Khandelwal et al., 2020; Guu et al., 2020; Karpukhin et al., 2020).

Despite their success, retrieval-augmented methods for the majority of non-knowledge-intensive (NKI) tasks remain under-explored. Following Lewis et al. (2020), we define tasks that "humans could not reasonably be expected to perform without access to an external knowledge source" as knowledge-intensive (KI) tasks and the others as NKI tasks. Previous studies (Karpukhin et al., 2020; Izacard and Grave, 2021; Izacard et al., 2022) have extensively explored the potential of retrieval-augmented methods for various KI tasks. As for NKI tasks, most efforts are devoted to language modeling (Khandelwal et al., 2020; Guu et al., 2020), text generation (Lewis et al., 2020), and machine translation (Zhang et al., 2018; Khandelwal et al., 2021), although there is a wide range of NKI tasks, such as sentiment analysis (Ding et al., 2008; Socher et al., 2013), text classification (Hovy et al., 2001; Li and Roth, 2002) and linguistic acceptability (Warstadt et al., 2018). Therefore, we ask this question: *Can retrieval-augmented methods assist on a wider range of NKI tasks?*

However, leveraging retrieval-augmented methods for more types of NKI tasks faces two major challenges. On the one hand, there is *a demand for diverse relevance score functions*. To retrieve the most desirable evidence from the index, proper relevance score functions are needed. Although the relevance score functions suitable for predicting the next token distribution are well-studied in works on language modeling, text generation, and ma-

[*] E-mail: guo-zc21@mails.tsinghua.edu.cn
[†] Corresponding authors. E-mail: lipeng@air.tsinghua.edu.cn, liuyang2011@tsinghua.edu.cn.

Findings of the Association for Computational Linguistics: ACL 2023, pages 10896–10912
July 9-14, 2023 ©2023 Association for Computational Linguistics

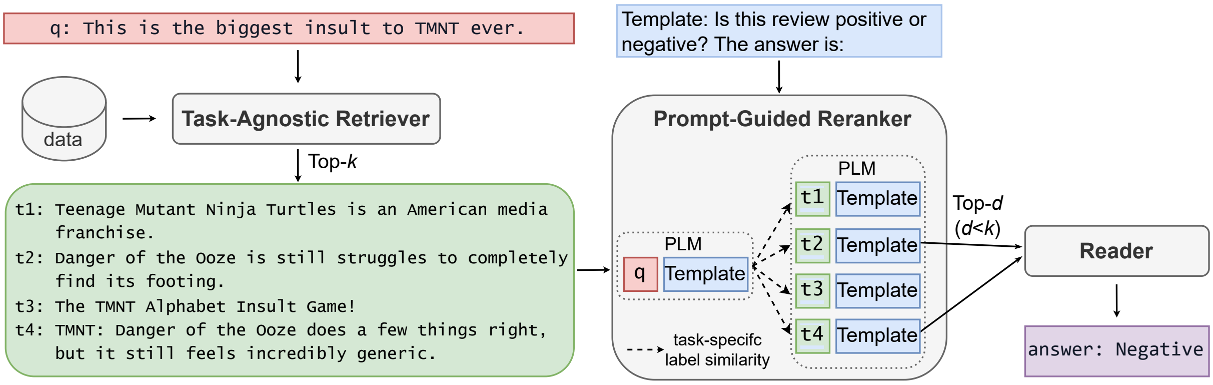

q: This is the biggest insult to TMNT ever.
data
Task-Agnostic Retriever
Top-k
t1: Teenage Mutant Ninja Turtles is an American media franchise.
t2: Danger of the Ooze is still struggles to completely find its footing.
t3: The TMNT Alphabet Insult Game!
t4: TMNT: Danger of the Ooze does a few things right, but it still feels incredibly generic.
Template: Is this review positive or negative? The answer is:
Prompt-Guided Reranker
PLM
q
Template
PLM
t1
Template
t2
Template
t3
Template
t4
Template
task-specifc label similarity
Top-d (d<k)
Reader
answer: Negative

Figure 1: The framework of our proposed Prompt-Guided Retrieval Augmentation (PGRA) method. We first retrieve candidates through a task-agnostic retriever (Section 2.1), then use a task-specific prompt and pre-trained language model (PLM) to rerank the candidates (Section 2.2). We send the top results to the reader to make predictions. (Section 2.3).

chine translation, NKI tasks require more diverse relevance score functions. For example, the text classification task may favor evidence with similar sentence-level semantics (Reimers and Gurevych, 2019; Gao et al., 2021b) while the linguistic acceptability task may prefer linguistically similar evidence (Warstadt et al., 2018). Therefore, it is non-trivial to satisfy all these diverse requirements in a single framework. On the other hand, there is *a dilemma between training cost and task performance*. The external knowledge index and retriever are crucial for the performance of a retrieval-augmented method (Lee et al., 2019; Karpukhin et al., 2020; Izacard and Grave, 2021). Previous works show that joint training index with the dense model results in better performance (Guu et al., 2020; Xiong et al., 2020). However, due to the large size of external knowledge, updating the index periodically during training is computationally expensive. On the contrary, keeping the index static is computationally cheap but makes it hard to meet the diverse requirements of NKI tasks. Therefore, it is difficult to balance the trade-off between training cost and task performance.

To address these challenges, we propose a two-stage framework, entitled PGRA, to better retrieve task-specific resources for NKI tasks. The overall framework is shown in Figure 1. In the first stage, we use a task-agnostic retriever to recall candidate evidence, which builds a shared static index for all tasks. In the second stage, we adopt prompt-guided pretrained language models (PLMs; Brown et al., 2020; Zhang et al., 2022) as a reranker to rerank the candidates according to the task-specific relevance score functions. Finally, we feed the reranked top evidence to the reader to generate answers. By leveraging textual prompts, our framework can satisfy the demand for diverse relevance score functions. As both the retriever and the reranker are training-free, the expensive computational cost of periodical index update in training is avoided. At the same time, experimental results justify the effectiveness of our framework on various datasets. Therefore, we successfully break the dilemma between training cost and task performance.

Our main contributions are three-fold:

- We propose a prompt-guided retrieval augmentation method for a wider range of non-knowledge-intensive tasks, which are hardly explored in previous works.
- By combining the retrieval-and-rerank procedure with textual prompts, our framework maintains reasonably low training cost while satisfying diverse task-specific relevance score function requirements.
- Extensive experimental results and analysis show that our framework is effective for diverse non-knowledge-intensive tasks.

2 Methods

In this section, we introduce our proposed Prompt-Guided Retrieval Augmentation (PGRA) method, as shown in Figure 1. Our proposed method mainly has three components: (i) a *task-agnostic retriever* using a shared retriever to build static indexes to select top-k candidate evidence from large-scale external resources; (ii) a *prompt-guided reranker* adopting PLMs to measure task-specific relevance for reranking candidate evidence; (iii) a *reader* taking the final top-d ($d < k$) ranked evidence as augmentations to generate answers.

2.1 Task-Agnostic Retriever

Given that the external resource is extremely large-scale, from millions (Khandelwal et al., 2020, 2021) to billions (Wang et al., 2022; Izacard and Grave, 2021; Chen et al., 2017), we use a shared retriever to build the static index once. The key and value of the index are the task-agnostic text representation and the text itself, respectively. The index will be shared across tasks, and thus we save a significant amount of training cost (See Section 4.5 for discussion). Formally, given the input as query q, and the external resource containing a bunch of text $\mathcal{R} = \{t_1, t_2, \cdots, t_{|\mathcal{R}|}\}$, we firstly encode representations for both query and text, which can be denoted as $\text{Enc}(q)$ and $\text{Enc}(t_i)$, respectively. The representations of text then serve as keys of the index. Then, we use a dense inner product to compute the similarity $\text{Sim}(q, t_i)$ based on the index:

$$\text{Sim}^{\text{agnostic}}(q, t_i) = \frac{\exp(\text{Enc}(q) \cdot \text{Enc}(t_i))}{\sum_{j=1}^{|\mathcal{R}|} \exp(\text{Enc}(q) \cdot \text{Enc}(t_j))}. \tag{1}$$

With the similarity scores, we get the top-k nearest evidence according to retrieval distribution which is the softmax over these scores. Then we follow the `faiss` (Johnson et al., 2019) implementation to efficiently complete the approximate retrieval via Maximum Inner Product Search (MIPS). These top-k pieces of evidence are regarded as candidates for the second stage for further reranking.

2.2 Prompt-Guided Reranker

As discussed above, the task-agnostic retriever in the first stage selects the nearest candidates by evaluating the similarity of the static indexes between input and external text. However, such shared retrievers neglect the fact that different NKI tasks prefer their own task-specific relevance score functions, which is crucial to retrieve useful evidence.

In order to meet the demand for diverse relevance score functions, we further design a task-specific reranker in the second stage. To avoid expensive calculations for training a task-specific retriever per NKI task, we exploit the in-context learning ability of prompt-guided PLMs.

Template: SST-2

/* Example */
Does the following sentence have a positive or negative sentiment?
one long string of cliches .
The answer is negative.

/* Test data */
Does the following sentence have a positive or negative sentiment?
the performances take the movie to a higher level .
The answer is

Table 1: The prompt instances of in-context learning in our prompt-guided retriever. We use 8 examples per prompt. More details can be found in Appendix I.

At first, we adopt in-context learning under the few-shot setups to encode task-specific representations of the input query q and the top-k pieces of evidence $\{e_1, e_2, \cdots, e_k\}$. Specifically, we design a prompt specialized for each task by concatenating m exemplars randomly sampled from the training datasets with manually written task descriptions as shown in Table 1. Then, we feed an auto-regressive PLM (*e.g.*, OPT; Zhang et al., 2022) with both constructed prompts and our input to obtain the task-specific representations of the next predicted tokens:

$$\begin{aligned} \text{prefix} &= p_1, l_1, p_2, l_2, \cdots, p_m, l_m \\ h_q^* &= \text{PLM}([\![\text{prefix}; p_q]\!]) \\ h_{e_i}^* &= \text{PLM}([\![\text{prefix}; p_{e_i}]\!]), \end{aligned} \tag{2}$$

where $p_1, p_2, \cdots, p_m$ are the m prompts of the examplars, $l_1, l_2, \cdots, l_m$ are the labels, p_q and p_{e_i} ($i = 1, \cdots, k$) are the prompts of the input query and evidence e_i, respectively. The prefix text is then concatenated to the prompts of the query or the evidence as the textual input. Lastly, the inputs are fed to the model to generate the last hidden states of the first new token $h_q^* \in \mathbb{R}^d$ and $h_{e_i}^* \in \mathbb{R}^d$.

It is worth noting that text in the external knowledge resource may lack explicit labels for NKI tasks. Through in-context learning with prompt guidance, the representations of the inputs and external evidence encoded by the PLM implicitly contain different critical features to solve various tasks. Similar to the first stage, we compute the similarity between the representations of input q and its candidate evidence e_i, which reflects their task-specific relevance:

$$\text{Sim}^{\text{task-specific}}(q, e_i) = \frac{\exp(h_q^* \cdot h_{e_i}^*)}{\sum_{j=1}^{k} \exp(h_q^* \cdot h_{e_j}^*)}. \tag{3}$$

Finally, we rerank the candidate evidence according to the aforementioned task-specific relevance score and select the top-d results for the reader in the next section.

2.3 Reader

To encode useful information from the reranked evidence and infer the final answer for the query text q, we use the FiD (Fusion-in-Decoder; Izacard and Grave, 2021) model as our reader, which has a Seq2seq pre-trained Transformer (Vaswani et al., 2017) such as T5 (Raffel et al., 2020). Specifically, each piece of evidence obtained from the reranker is concatenated with the query, which is independently fed into the encoder. The decoder takes the embeddings of these concatenations produced by the encoder and computes cross attention over them to give the final answer prediction. Following the prompt-based learning (Schick and Schütze, 2021; Liu et al., 2021), we transfer the NKI tasks to the form of language modeling, where the answers are deduced according to the label prediction in a context. The overall reader is trainable and the parameters are updated given the training samples of the required NKI tasks.

3 Experiments

3.1 Experimental Setups

Tasks and Metrics. Following the setups in LM-BFF (Gao et al., 2021a), We conduct the experiments mainly on four types of NKI tasks: (1) Sentiment analysis. We use a various of datasets from different domains, including SST-2 (Socher et al., 2013) and SST-5 (Socher et al., 2013) for the general domain with two and five labels, CR (Ding et al., 2008) for comment reviews, MR (Pang and Lee, 2004) for movie reviews, MPQA (Wiebe et al., 2005) for news opinions; (2) Linguistic acceptability. We adopt CoLA (Warstadt et al., 2018), which aims to discriminate whether a sentence is grammatically correct; (3) Question classification. We use TREC (Hovy et al., 2001; Li and Roth, 2002), in which a question needs to be classified into six categories; (4) Subjectivity analysis. We use Subj (Pang and Lee, 2004), which has to judge whether the sentence is subjective or objective. As for metrics, we report Matthew's correlation for CoLA while reporting accuracy in all other tasks. More details about datasets and metrics can be found in Appendix G.

External Resources and Models. As for the external resources, we use Wiki1M following Gao et al. (2021b). Furthermore, in the first stage, we use BERT-base-uncased (Devlin et al., 2019) as our shared task-agnostic retriever. We also compare with other retrievers of the first stage in Section 4.6. In the second stage, we use OPT-13b (Zhang et al., 2022) as our auto-regressive PLMs to obtain the task-specific representations. We further explore the effects on the size of our PLMs in Section 4.3. Finally, we adopt T5-base and T5-large (Raffel et al., 2020) as our readers to generate answers.

Implementation Details. We use the checkpoints of T5-base, T5-large, and OPT-13b from HuggingFace[1]. Our manually designed prompts are obtained from PromptSource (Bach et al., 2022). We finetune the T5 model on each task with the AdamW (Loshchilov and Hutter, 2019) optimizer. We search hyper-parameters of learning rate of {1e-5, 2e-5, 5e-5, 8e-5, 1e-4} and batch sizes of {4, 8}. We set the number of top-k in the first stage to 150, while the number of top-d in the second stage is 16 with T5-base and 8 with T5-large due to computational resource limitation. We further compare the effect of k and d in Section 4.2. We use 8 shots for prompts during reranking in the second stage. Our experiments are conducted with one NVIDIA V100 GPU.

Baselines. We compare our proposed method PGRA with the following baselines: (1) In-context learning (ICL; Brown et al., 2020), which directly uses OPT-13b, the same as our PLM in the second stage, to generate answers under the few-shot setups (8 shots in our settings); (2) T5 (Raffel et al., 2020), which use T5-base and T5-large in supervised learning; (3) k-Nearest Neighbour (k-NN; Cunningham and Delany, 2020), in which the model makes a majority vote based on distances between embeddings; (4) LM-BFF (Gao et al., 2021a), which is a few-shot inference method tuned with dedicated prompts; (5) RAG (Lewis et al., 2020), which treats context samples as hidden variables and jointly trains the retriever and generator; (6) FiD (Izacard and Grave, 2021), which concatenates query and context samples in the encoder and generates answers with cross attention. To ensure a fair comparison, we uniformly adopt the same reader (*i.e.*, T5-base and T5-large) for retrieval-augmented methods. As for k-NN and

[1] https://huggingface.co/models

Method	Retrieval	SST-2	SST-5	CoLA	TREC	CR	MR	MPQA	Subj	Average
ICL (OPT-13b)	✗	93.0	46.0	1.8	26.8	73.2	61.7	71.6	51.1	53.2
T5-base **(220M)**										
k-NN	✓	59.2	22.8	1.0	28.0	51.8	55.1	52.6	72.1	42.9
LM-BFF	✗	86.0	45.5	5.5	76.2	90.0	83.1	82.3	90.2	69.9
T5-base	✗	91.3	56.7	30.4	80.4	89.8	89.4	89.2	96.0	77.9
RAG	✓	93.0	**57.5**	**58.5**	80.4	87.2	90.2	89.5	96.5	81.6
FiD	✓	92.2	56.6	56.9	80.8	91.3	90.1	89.8	96.6	81.8
PGRA (Ours)	✓	**93.9**	56.9	57.0	**80.8**	**91.7**	**91.1**	**90.3**	**97.0**	**82.3**
T5-large **(770M)**										
k-NN	✓	64.5	23.6	2.1	28.8	56.8	58.2	53.7	72.4	45.0
LM-BFF	✗	90.8	49.0	6.9	70.6	91.1	83.5	89.5	88.4	71.2
T5-large	✗	95.2	59.2	60.7	80.8	92.1	91.5	**90.7**	97.3	83.4
RAG	✓	95.2	57.2	60.1	80.2	91.2	92.1	90.6	96.4	82.9
FiD	✓	94.8	59.5	60.2	80.8	92.4	**92.5**	90.6	**97.5**	83.5
PGRA (Ours)	✓	**95.7**	**59.8**	**61.1**	**80.9**	**92.6**	92.4	90.6	**97.5**	**83.8**

Table 2: The results of baselines and our PGRA. For models with T5-base backbone, we use $d = 16$. For models with T5-large backbone, we use $d = 8$ in the second stage due to GPU memory limitation. The best results are **bolded**, and the second-best ones are underlined.

LM-BFF, we also use T5-base and T5-large for building representations and training. In the baseline of in-context learning, we use the same templates as ours in the second stage.

3.2 Results

We compare our proposed PGRA with the aforementioned baseline methods, where the results are shown in Table 2. We include results on both T5-base and T5-large models for generality reasons. We run our experiments three times and report details of each run in Appendix A. We report average results here and first-run results in the analysis section below.

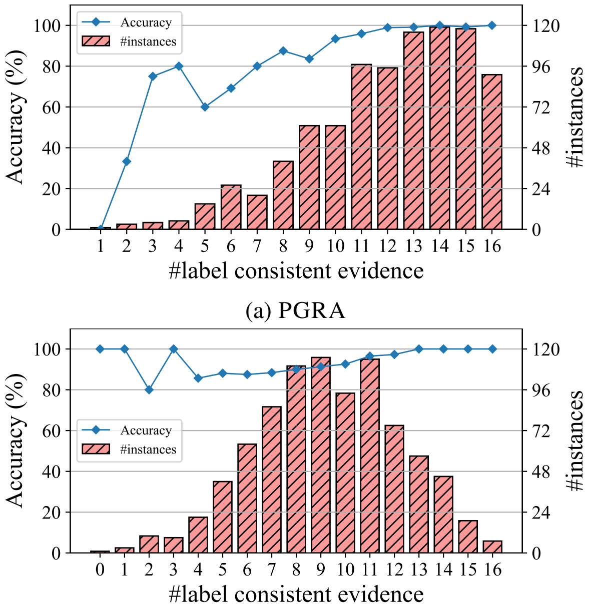
Firstly, the PGRA can significantly outperform the simple k-Nearest Neighbour and few-shot methods, including in-context learning with OPT-13b and LM-BFF. As for the k-Nearest Neighbour, it is simply based on the distances of embeddings encoded by T5. As for the few-shot methods, in-context learning uses prompts to elicit PLMs to generate answers without updating parameters. It is worth noting that we use in-context learning with OPT-13b as our prompt-guided reranker in the second stage. The performance of in-context learning is ordinary, so it is surprising that it can assist on PGRA. We will further discuss the reason behind this in Section 4.1. Meanwhile, LM-BFF is further fine-tuned on the prompts to give answers. Thus, its performance is obviously higher than k-Nearest Neighbour and in-context learning with OPT-13b but remains a large gap to PGRA.

Secondly, compared to supervised learning (*i.e.*, T5-base and T5-large) and retrieval-augmented baselines, PGRA still outperforms them across most tasks. Specifically, the line of retrieval methods with a T5-base reader outperforms supervised learning with the T5-base model, while retrieval-augmented methods with a T5-large reader are worse or comparable to supervised learning with the T5-large model. Furthermore, our method PGRA can obviously surpass these baselines, in both T5-base and T5-large setups. In conclusion, extensive experimental results have shown that our PGRA is effective on diverse NKI tasks.

4 Analysis

4.1 Effects of Label Consistency

In this section, we probe the influence of retrieved evidence on the model performance of our PGRA from the aspect of label consistency. Note that our external text is without any task-specific labels. Therefore, we use a T5-base model fine-tuned on the specific task, which is the closest to our PGRA reader but without retrieval, to generate *pseudo-label* for all text in the external resource. In detail, if the pseudo-label of evidence is the same as the ground-truth label of the input, we say the evidence is *consistent* with the input. We can then directly detect the relation between the number of consistent evidence and model performance at the instance level. Specifically, out of 16 pieces of total retrieved evidence, the number of consistent evidence with the same (pseudo) labels as the input



(a) PGRA

(b) FiD

Figure 2: The pseudo label consistency of samples in SST-2 with PGRA and FiD (T5-base models for both). We plot the accuracy scores of instances with different numbers of label-consistent evidence, along with the number of such instances.

varies from 0 and 16.

Taking the SST-2 task as an example, we count the total number of instances with different numbers of consistent evidence. We then compute the average accuracy of PGRA for the instances with the same number of consistent evidence. The results are shown in Figure 2a. Firstly, since we rerank the evidence based on the relevance score of pseudo-labels, the number of instances also rises as the number of consistent evidence increases. The phenomenon indicates that we can always find sufficient task-specific evidence retrieved from the first stage, except for a small part of inputs which is possibly caused by the limitation of the k's size in the first stage. Secondly, the average accuracy is also rising as the number of consistent evidence increases, which reflects that the model performance is related to the (pseudo) label consistency. However, when the number of consistent evidence is small (*i.e.*, 3 and 4), the accuracy can also be high. This is because the number of instances is too small, so the result is insignificant. Furthermore, it is interesting to find that when the number of consistent evidence is high enough (*i.e.*, larger than 13), the accuracy approaches 100%, which shows that there exists high potential in increasing label consistency to improve model performance.

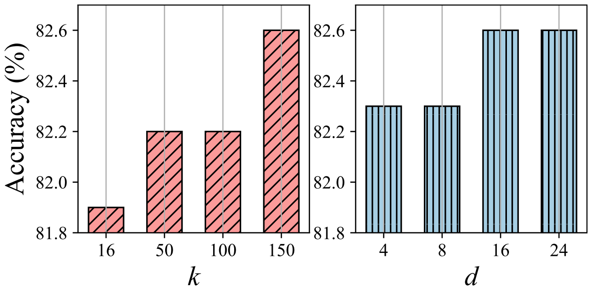


Figure 3: Accuracy against k (left) and d (right). Details of performance on different tasks can be found in Appendix C.

We use the same method to plot the label consistency figure on the FiD baseline, shown in Figure 2b. As can be seen from the figure, it still holds that the more label-consistent evidence, the higher accuracy the model can achieve. The difference between PGRA and FiD is that PGRA can retrieve more label-consistent evidence than FiD.

4.2 Effects of k and d

In this section, we further investigate the effects of k and d on the performance, where k and d are the numbers of final retrieved evidence in the first and second stages, respectively. In detail, we run PGRA with different k or different d, while other setups keep the same as main experiments. As seen from Figure 3, larger k values can consistently improve the average performance, while larger d values maintain a relatively stable trend. As for k, larger k values mean providing more candidate evidence for the second stage reranker to find more appropriate instances with (pseudo) label consistency. As for d, larger d values indicate more consistent evidence if the proportion of consistent evidence keeps the same. At the same time, their top consistent evidence is the same, and the candidate evidence is fixed with the same k, so their performance is close. In our expectation, the PGRA can better solve diverse NKI tasks with larger k if enough computing resources are allowed.

4.3 Effects of OPT Model Sizes

In this section, we first investigate the effects on the performance of different sizes of the OPT models used in the prompt-guided reranker. Specifically, we vary the size of OPT models and conduct experiments in five downstream tasks. The model performances are shown in the orange line of Figure 4. The overall trend is obviously that the larger OPT models can achieve better performance.

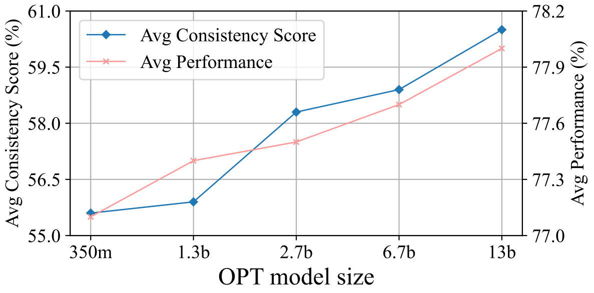


Figure 4: Average performance and average consistency score on 5 tasks (SST-2, SST-5, CoLA, MR and MPQA) against different OPT model sizes. Detailed information can be found in Appendix C.

We believe that the larger OPT models have better abilities to apply task-specific features to encode representations, and further obtain more effective task-specific relevance scores to retrieve evidence.

To validate this assumption, we further investigate the relations between (pseudo) label consistency and model performance of different OPT model sizes. We define the pseudo-label consistency score (*i.e.*, *consistency score*) as the proportion of retrieved instances with the same pseudo-label as the input. For example, given the input with a positive ground-truth label, when our PGRA recalls 5 consistent and 3 inconsistent pieces of evidence, the consistency score is $5/8 = 62.5\%$. As shown in the Figure 4, overall, larger models with higher consistency scores result in better performance, which is within expectation.

4.4 Effects of Evidence Granularity

In this work, we propose to use a task-specific relevance score to retrieve from sentence-level external resources, rather than popular passage-level used in previous studies (Chen et al., 2017; Izacard and Grave, 2021; Guu et al., 2020). To demonstrate that our granularity of external evidence is appropriate, we compare the model performance between sentence-level and passage-level evidence. As for passage-level evidence, we use WikiDPR (Chen et al., 2017) as external resources. We randomly sample 1M passages from WikiDPR to keep the same data size as our sentence-level external resource in the main experiment. The results are shown in Figure 5. Across all NKI tasks, our sentence-level setup performances significantly surpass passage-level setup. This phenomenon indicates that sentence-level evidence can better satisfies the task-specific demands for NKI tasks. For example, it is easier to show a clear sentiment orientation in a sentence than in a paragraph.

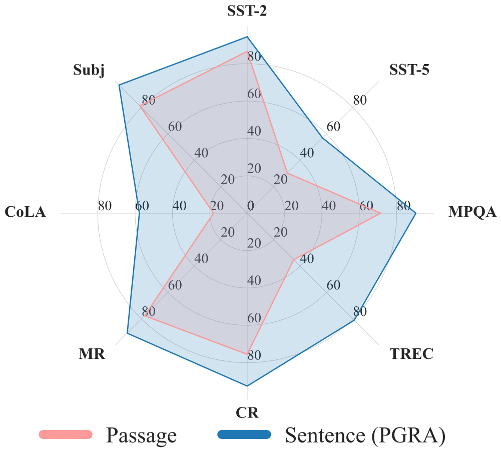


Figure 5: Performance of PGRA with passage-level and sentence-level external datastores.

4.5 Training Cost

To solve the dilemma between training cost and task performance, we propose PGRA where both the retriever and reranker are training-free. To demonstrate this statement, in this section, we approximately compare the training cost of our method PGRA with training a task-specific retriever per task, the latter of which needs periodical refreshing indexes (*i.e.*, refreshed-index models). Considering a significant amount of training time concentrates on building and refreshing indexes, we mainly statistic this part. Due to the limitation of computation resources, we conduct our main experiment on 1M data from Wikipedia. In our PGRA, we only need to build the index once without extra training, and the time cost c is about 0.5 hours. However, although the time cost c of building index is almost the same, they need to periodically refresh the index n times to learn a task-specific retriever. Lastly, for all h tasks, their total training cost is $c \times n \times h$, which is much larger than our time cost c. It is worth noting that the external resource is usually much larger than ours (Chen et al., 2017; Wang et al., 2022; Izacard et al., 2022), so the gap between refreshed-index models and our PGRA will further grow to explode.

4.6 Generalization on Retrievers

In this section, we study the generalization of PGRA with different first-stage retrievers. We use popular retrievers like BM25 (Robertson and Zaragoza, 2009), BERT (Devlin et al., 2019), SimCSE (Gao et al., 2021b) to compare FiD and our

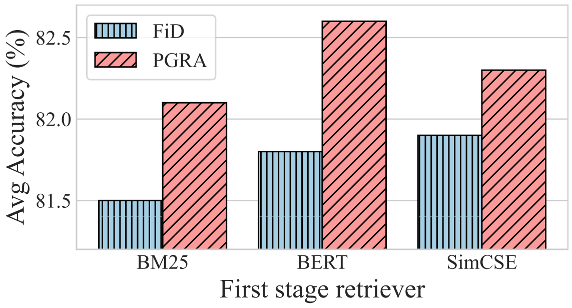


Figure 6: Comparison between FiD and our PGRA with BM25, BERT and SimCSE retrievers. More details of specific performance in all tasks can be found in Appendix E.

PGRA. As shown in Figure 6, our method PGRA consistently outperforms FiD, no matter which retriever to use. This phenomenon indicates that PGRA can adapt to different types of retrievers in the first stage to solve various NKI tasks. Furthermore, the retriever with BM25 performs worse than both BERT and SimCSE counterparts, which is consistent with previous studies (Zhao et al., 2022).

4.7 Case Study

In Table 3, we present a case in SST-2 with different retrieved evidence from baselines (*i.e.*, FiD and RAG) and our PGRA. As shown in the table, our PGRA can exactly predict the correct answer, while both FiD and RAG are wrong. To further analyze the retrieved evidence from different methods, we find that sentences retrieved by FiD and RAG may have overlapped tokens or similar semantics. For example, the retrieved evidence from FiD is highly related to filming and stories, consistent with the "title", "characters", and "camera" in the input. But their retrieved evidence hardly has the same sentiment orientation to assist the downstream task. Some of them may have even opposite sentiments, such as the second sentence retrieved by FiD. However, our retrieved evidence from PGRA clearly has a negative sentiment orientation, though some may not have explicitly relatedness with the query, such as the second retrieved sentence. In general, evidence retrieved by our PGRA method based on task-specific relevance can effectively improve performance on NKI tasks.

5 Related Work

Retrieval-augmented methods are widely used for knowledge-intensive tasks such as question answering (Chen et al., 2017; Karpukhin et al., 2020; Izacard and Grave, 2021; Izacard et al., 2022), where explicit knowledge is required to achieve reasonable performance, even for human (Lewis et al., 2020). Such systems usually follow a *retriever-reader* architecture, where an existing retriever like BM25 (Robertson and Zaragoza, 2009) or a trained dual-encoder (Lee et al., 2019; Luan et al., 2021) is used, followed by a reader model to fuse the retrieved results. We focus on non-knowledge-intensive tasks and propose a prompt-guided retrieval method for mining fine-grained textual information across multiple tasks, without training a specific retriever for each of them. Recently, Wang et al. (2022) also applied retrieval-augmented methods to more general tasks by keeping a shared BM25 retriever unchanged for each task while modifying the reader for information filtering. In contrast, we propose a two-stage retrieval method to find task-specific information at a low cost for different downstream tasks.

Dataset: SST-2	
Input	The title not only describes its main characters but the lazy people behind the camera as well.
Label	*Negative*
Method: FiD	
Prediction	*Positive*
Evidence	(1) The story overlaps science fiction, theology, and philosophy. (2) However, the film's greatness is not limited to a few isolated scenes.
Method: RAG	
Prediction	*Positive*
Evidence	(1) The 1978 King Cup was the 20th season of the knockout competition since its establishment in 1956. (2) Per Kristian Norvik was born in Vadsø, Norway on February 10, 1938.
Method: PGRA	
Prediction	*Negative*
Evidence	(1) Once it had been shown that the film could not be realized, "The Works" was officially abandoned. (2) The play can also be seen as a discussion of romanticism and reality, in a quite disillusional way.

Table 3: Case study of FiD, RAG, and our PGRA with the top-2 retrieved evidence in SST-2.

Prompt-based methods gained much advance in recent years (Schick and Schütze, 2021; Liu et al., 2021; Gao et al., 2021a), where downstream tasks can be solved via transforming the problem to the form of language modelling. Combined with PLMs such as GPT-3 (Brown et al., 2020) and OPT (Zhang et al., 2022), such methods show strong performance under zero-shot or few-shot settings. Recently, there are also some works that leverage prompts for retrieval. For example, Asai et al. (2022) collected large-scale instructionannotated datasets for training instruction-guided retrievers for tasks, van de Kar et al. (2022) use prompts for searching regex-based patterns from unlabeled corpora. Our method is inspired by these works and different in that we leverage the pre-trained models for retrieving according to task-specific relevance and propose an efficient retrieval-augmented method for NKI tasks.

6 Conclusion

In this paper, considering the demand for diverse relevance score functions to solve wider NKI tasks, we propose a two-stage method PGRA. In the first stage, we use a task-agnostic retriever for building shared static indexes to select candidate evidence. In the second stage, we design a prompt-guided reranker to rerank candidates with task-specific relevance for the reader. Extensive experimental results show that our proposed method PGRA can overall outperform previous state-of-the-art retrieval-augmented methods. Furthermore, we explore the influence of label consistency between input and retrieved evidence from the prompt-guided reranker and demonstrate the generality of our PGRA on both evidence granularities and types of retrievers. In the future, we will consider ways to improve the pseudo-label consistency to enhance model performances according to our analyses.

Limitations

In this work, we present PGRA to retrieve task-specific context evidence to support NKI tasks. However, our work has some limitations. Firstly, we have not experimented with our PGRA on sentence-pair tasks, such as MRPC (Dolan and Brockett, 2005), in which the model needs to infer the relationship between two sentences. Retrieving two sentences from an external datastore is non-trivial as there are hardly sentence pairs in the Wikipedia datastore. A larger corpus with more diverse data sources may help in this case. Secondly, We restrict our PGRA to classification tasks but not generation tasks. Similar to sentence-pair tasks, retrieving sentences that may help the model generate text is more complex. For example, data related to both the source and the target may help in machine translation (Khandelwal et al., 2021). We will research this question in the future. Last but not least, we have not extensively tested the performance of our method on KI tasks, except for some preliminary analysis in Appendix F. This restricts the generality of our methods. Solving KI tasks depends on knowledge in the passage-level external datastore while matching such information needs possibly more specialized prompts for our method. Thus, it is for our future work.

Ethics Statement

Currently, large language models with retrieval augmentation require a large amount of computation in indexing a large-scale datastore, retrieving from that large datastore and refreshing index during training. Despite improving model performance, the retrieval augmentation methods need too much computation power. This not only limits the usability of such models but also harms the fairness in this community. Our work tries to balance the performance of retrieval augmentation methods and the training cost, in that our method does not need to retrain a new retriever and rebuild an index when facing a new task. This may help the community in developing new low-cost methods.

During selecting the external datastore and tasks, we follow previous studies and choose the well-known Wikipedia dataset and common tasks. Biases from the data may be reflected in the results. In addition, when using the model on a larger scale, more consideration needs to be paid to deal with biases in retrieved text.

Acknowledgements

This work is supported by the National Key R&D Program of China (2022ZD0160502) and the National Natural Science Foundation of China (No. 61925601, 62276152, 62236011). We also want to show our gratitude to Ziyue Wang for her suggestions on the writing of the paper.

References

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260*.

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Xiangru Tang, Mike Tian-Jian Jiang, and Alexander M.

Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proc. of ICML*.

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. *Proc. of NeurIPS*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proc. of ACL*.

Padraig Cunningham and Sarah Jane Delany. 2020. k-nearest neighbour classifiers: 2nd edition (with python examples). *CoRR*, abs/2004.04523.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proc. of WSDM*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proc. of IWP*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proc. of ACL*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Proc. of EMNLP*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proc. of ICML*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proc. of HLT*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proc. of EACL*.

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*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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 *Proc. of EMNLP*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proc. of ICLR*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *Proc. of ICLR*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proc. of ACL*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proc. of NeurIPS*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proc. of COLING*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proc. of ICLR*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of ACL*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of EMNLP*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proc. of EACL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proc. of ACL*.

Mozes van de Kar, Mengzhou Xia, Danqi Chen, and Mikel Artetxe. 2022. Don't prompt, search! Mining-based zero-shot learning with language models. *arXiv preprint arXiv:2210.14803*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NeurIPS*.

Zhenhailong Wang, Xiaoman Pan, Dian Yu, Dong Yu, Jianshu Chen, and Heng Ji. 2022. Zemi: Learning zero-shot semi-parametric language models from multiple tasks. *arXiv preprint arXiv:2210.00185*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. In *Proc. of NAACL*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pre-trained language models: A survey. *arXiv preprint arXiv:2211.14876*.

A Multiple Runs of the Main Experiment

We run PGRA three times under the settings of the main experiment in Table 2 and report results of these runs in Table 8.

B Experiments with More Retrieved Evidence For FiD baselines

We run additional experiments for FiD (T5-base) baseline with more retrieved evidence. The results are shown in Table 4. It can be seen that with more retrieved evidence, although the average scores become higher, FiD still underperforms PGRA.

Datasets	$d = 8$	$d = 48$
SST-2	92.2	93.3
SST-5	56.6	56.8
CoLA	56.9	56.3
TREC	80.8	81.0
CR	91.3	91.2
MR	90.1	90.8
MPQA	89.8	89.5
Subj	96.6	96.8
Avg.	81.8	82.0

Table 4: Detailed analysis of the impact of top-d in the second stage.

C Impact of k, d and OPT Model Sizes

We explore the impact of k, d and second-stage OPT model sizes. The full analysis is shown in Section 4.2 and Section 4.3. Table 5, Table 6 and Table 9 show detailed performance of our method in each task. For each ablation, we keep other hyper-parameters the same as used in Table 2.

Datasets	$k = 16$	$k = 50$	$k = 100$	$k = 150$
SST-2	93.6	92.8	93.7	94.4
SST-5	55.7	56.4	56.0	57.0
CoLA	56.0	58.9	56.8	57.7
TREC	80.6	80.6	80.6	81.0
CR	91.7	91.8	92.1	92.5
MR	90.8	90.5	91.5	90.8
MPQA	90.1	90.2	89.8	90.3
Subj	97.0	96.7	97.0	97.0
Avg.	81.9	82.2	82.2	82.6

Table 5: Detailed analysis of the impact of top-k in the first stage.

D Label Consistency

We include the details of label consistency scores of our PGRA with different second-stage OPT models on each task in Table 10.

Datasets	$d = 4$	$d = 8$	$d = 16$	$d = 24$
SST-2	94.4	94.3	94.4	94.2
SST-5	56.2	56.5	57.0	57.8
CoLA	57.3	55.7	57.7	56.4
TREC	80.8	80.6	81.0	80.8
CR	91.7	92.2	92.5	92.1
MR	90.9	91.5	90.8	91.8
MPQA	90.1	90.4	90.3	90.5
Subj	96.7	97.0	97.0	97.0
Avg.	82.3	82.3	82.6	82.6

Table 6: Detailed analysis of the impact of top-d in the second stage.

Datasets	learning rate	batch size
SST-2	8e-5	8
SST-5	2e-5	8
CoLA	8e-5	8
TREC	8e-5	4
CR	1e-5	8
MR	1e-4	8
MPQA	8e-5	4
Subj	8e-5	8

Table 7: Information of the tasks.

E Generality on Retrievers

We include the detailed performance of FiD and our PGRA on all tasks with different first-stage encoders, namely BM25, BERT and SimCSE. The results are shown in Table 11.

F Generalization on KI tasks

We perform experiments on the FEVER (Thorne et al., 2018) benchmark. FEVER is a fact verification task, requiring a model to classify whether a claim is factually correct. Due to resource limitations, we sample 5k claim-label pairs from the training set and 1k pairs from the validation set. We run FiD and PGRA with both T5-base backbone and keep other hyperparameters the same as in Table 2. Note that we did this experiment with a sentence-level datastore (Wiki1m). FiD and PGRA achieve 73.8% and 77.7% accuracy respectively. The results confirm again the performance increase with PGRA. However, one might notice that the performance of FiD with a traditional passage-level datastore can achieve better performance. We acknowledge this as a limitation of our method because a passage-level datastore requires much different relevance metrics as stated in the Limitation section. This is also a possible future direction.

Runs	SST-2	SST-5	CoLA	TREC	CR	MR	MPQA	Subj	Average
				T5-base **(220M)**					
Run 1	94.4	57.0	57.7	81.0	92.5	90.8	90.3	97.0	82.6
Run 2	93.5	56.5	57.6	80.8	90.6	91.3	90.1	97.0	82.2
Run 3	93.9	57.1	57.5	80.6	91.9	91.2	90.4	97.0	82.5
Average	93.9	56.9	57.6	80.8	91.7	91.1	90.3	97.0	82.4
Std	0.45	0.32	0.10	0.20	0.97	0.26	0.15	0.0	0.21
				T5-large **(770M)**					
Run 1	96.0	59.4	64.0	81.2	92.8	93.0	90.8	97.6	84.4
Run 2	95.4	60.1	60.1	81.0	92.0	92.3	90.5	97.3	83.6
Run 3	95.8	60.0	59.1	80.6	92.9	91.9	90.5	97.6	83.6
Average	95.7	59.8	61.1	80.9	92.6	92.4	90.6	97.5	83.8
Std	0.31	0.38	2.59	0.31	0.49	0.56	0.17	0.17	0.45

Table 8: Multiple run results of PGRA.

G Datasets and Metrics

We use the Wiki1M from SimCSE (Gao et al., 2021b) as our external datastore. This dataset is a subset of Wikipedia and used in (Gao et al., 2021b). We report information on tasks in Table 12. We use the same configuration as (Gao et al., 2021a), including dataset splits.

H Training Details

As stated in Section 3.1, we search hyper-parameters of learning rate of {1e-5, 2e-5, 5e-5, 8e-5, 1e-4} and batch sizes of {4, 8}. We train our models for 5000 steps on the training set. The best hyperparamters found are shown in Table 7.

I Prompts

We include all prompts used in all 8 tasks in Table 13.

Datasets	**SST-2**	**SST-5**	**CoLA**	**TREC**	**CR**	**MR**	**MPQA**	**Subj**	**Average**
OPT 350m	93.9	56.2	55.1	80.8	90.8	90.3	90.0	97.2	81.8
OPT 1.3b	93.8	57.8	54.8	80.6	92.4	90.5	90.0	96.7	82.1
OPT 2.7b	94.2	56.5	55.4	81.0	91.6	90.7	90.9	96.7	82.1
OPT 6.7b	93.6	57.5	56.2	80.6	92.2	90.6	90.6	96.8	82.3
OPT 13b	94.4	57.0	57.7	81.0	92.5	90.8	90.3	97.0	82.6

Table 9: Detailed analysis of the impact of OPT sizes with $k = 150, d = 16$.

Encoder	**SST-2**	**SST-5**	**CoLA**	**MR**	**MPQA**	**Average**	**Performance**
OPT 350m	63.4	28.8	69.0	60.3	56.6	55.6	77.1
OPT 1.3b	68.3	32.2	69.3	55.6	53.6	55.9	77.4
OPT 2.7b	68.9	32.5	69.0	65.8	55.2	58.3	77.5
OPT 6.7b	70.1	32.0	69.1	69.3	54.2	58.9	77.7
OPT 13b	75.2	30.2	69.1	70.5	57.3	60.5	78.0

Table 10: Pseudo-label consistency with different OPT models. “Average” is the average label consistency score on the five tasks. Performance is the average of the 5 tasks. We keep $k = 150, d = 16$ in this experiment.

Method	**SST-2**	**SST-5**	**CoLA**	**TREC**	**CR**	**MR**	**MPQA**	**Subj**	**Average**
FiD (BM25)	93.6	56.4	56.7	80.6	90.2	89.7	88.2	96.8	81.5
PGRA (BM25)	93.9	56.6	56.9	80.4	91.4	90.4	90.5	96.5	82.1
FiD (BERT)	92.2	56.6	56.9	80.8	91.3	90.1	89.8	96.6	81.8
PGRA (BERT)	94.4	57.0	57.7	81.0	92.5	90.8	90.3	97.0	82.6
FiD (SimCSE)	93.2	56.8	56.2	81.0	90.9	90.5	90.1	96.3	81.9
PGRA (SimCSE)	93.5	57.7	55.9	81.0	92.0	91.0	90.4	96.9	82.3

Table 11: Table of generalization performance of FiD and PGRA with different first-stage encoders.

Datasets	Type	Labels	Avg Input length
SST-2	Sentiment analysis	positive, negative	19
SST-5	Sentiment analysis	v. pos., positive, neutral, negative, v. neg.	18
CoLA	Linguistic acceptability	acceptable, unacceptable	8
TREC	Question classification	abbr., entity, description, human, loc., num.	10
CR	Sentiment analysis	positive, negative	19
MR	Sentiment analysis	positive, negative	20
MPQA	Sentiment analysis	positive, negative	3
Subj	Subjectivity analysis	subjective, objective	23

Table 12: Information of the tasks and datasets.

Template: SST-2/CR/MR/MPQA
/* Example */ Does the following sentence have a positive or negative sentiment? one long string of cliches . The answer is negative. /* Test data */ Does the following sentence have a positive or negative sentiment? the performances take the movie to a higher level . The answer is
Template: SST-5
/* Example */ What sentiment does this sentence have? terrible, bad, okay, good or great "with a romantic comedy plotline straight from the ages, this cinderella story doesn't have a single surprise up its sleeve ." The answer is bad /* Test data */ What sentiment does this sentence have? terrible, bad, okay, good or great hardly a film that comes along every day. The answer is
Template: CoLA
/* Example */ The following sentence is either "acceptable", meaning it is grammatically correct and makes sense, or "unacceptable". Which is it? I ordered if John drink his beer. The answer is unacceptable /* Test data */ The following sentence is either "acceptable", meaning it is grammatically correct and makes sense, or "unacceptable". Which is it? Angela characterized Shelly as a lifesaver. The answer is
Template: Subj
/* Example */ Is this a subjective or objective description? when the skittish emma finds blood on her pillow why does she still stay behind? The answer is objective /* Test data */ Is this a subjective or objective description? "at the end of the worst day of his life, bruce angrily ridicules and rages against god and god responds ." The answer is
Template: TREC
/* Example */ Which category best describes the following question: How far is it from Denver to Aspen. Choose from the following list: Description, Entity, Abbreviation, Person, Quantity, Location. The answer is Quantity. /* Test data */ Which category best describes the following question: What were Ottoman objectives? Choose from the following list: Description, Entity, Abbreviation, Person, Quantity, Location. The answer is

Table 13: The prompt instances of in-context learning in our prompt-guided reranker.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The Limitations section

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In section 1 Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

All sections

☑ B1. Did you cite the creators of artifacts you used?
Reference section

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3 experiments and all datasets are open

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 2 Method and Section3 Experiments

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 3 and Ethical statement

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix D

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix D & E and Section 3.1& 3.2

C ☑ Did you run computational experiments?

Section 3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3.1 & 3.2 & 4.5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.1 Appendix E

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3 and Appendix E

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 2 & Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.