Learning to Retrieve In-Context Examples for Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated their ability to learn in-context, allowing them to perform various tasks based on a few input-output examples. However, the effectiveness of in-context learning is heavily reliant on the quality of the selected examples. In this paper, we propose a novel framework to iteratively train dense retrievers that can identify high-quality in-context examples for LLMs. Our framework initially trains a reward model based on LLM feedback to evaluate the quality of candidate examples, followed by knowledge distillation to train a bi-encoder based dense retriever. Our experiments on a suite of 30 tasks demonstrate that our framework significantly enhances in-context learning performance. Furthermore, we show the generalization ability of our framework to unseen tasks during training. An in-depth analysis reveals that our model improves performance by retrieving examples with similar patterns, and the gains are consistent across LLMs of varying sizes.

1 Introduction

In-context learning (ICL) (Brown et al., 2020) is an emerging learning paradigm that allows LLMs to perform tasks with few-shot examples, without requiring any updates to the model parameters. This approach stands in stark contrast to traditional machine learning, where models are typically trained on large datasets of labeled examples (Devlin et al., 2019). In-context learning offers a significant advantage in domains where labeled data is scarce or expensive to obtain, as it greatly reduces the amount of required labeled data.

There are several challenges associated with understanding and enhancing the effectiveness of in-context learning. One such challenge is that LLMs can be highly sensitive to the quality of the in-context examples provided (Liu et al., 2022; Min et al., 2022). If the examples are not representative of the target task, then the model may not be able to learn effectively. Empirical studies (Liu et al., 2022; Luo et al., 2023) have demonstrated that using BM25 algorithm or off-the-shelf sentence embeddings (Reimers and Gurevych, 2019) to retrieve examples from the training set can substantially enhance the performance of in-context learning over random selection. Another approach involves training dense retrievers based on the feedback signals from LLMs, which has shown promising results in semantic parsing (Rubin et al., 2022), cross-task prompt retrieval (Cheng et al., 2023), and unified multi-task retrieval (Li et al., 2023). However, existing methods either focus on a relatively small language model (Rubin et al., 2022), or fail to exploit the fine-grained feedback information from LLMs in a principled manner (Li et al., 2023).

In this paper, we propose a novel framework, LLM-R (**LLM R**etriever), which aims to retrieve high-quality in-context examples for large language models. Given an initial set of retrieved candidates, our framework ranks them based on the conditional LLM log probabilities of the ground-truth outputs. Subsequently, a cross-encoder based reward model is trained to capture the fine-grained ranking signals from LLMs. Finally, a bi-encoder based dense retriever is trained using knowledge distillation. The reward model plays a crucial role in providing more informative soft-labels that are suitable for distillation, instead of using heuristically constructed one-hot labels. This pipeline can be iterated multiple times by retrieving a new set of candidates based on the latest dense retriever.

For evaluation purposes, we assemble a diverse set of 30 NLP tasks, which span 9 categories, including question answering, natural language inference, commonsense reasoning, and summarization, among others. Experimental results obtained using LLaMA-7B (Touvron et al., 2023) demonstrate that our model improves the in-context learning performance by an average of 7.8% compared to

random selection. Similar improvements are also observed on held-out tasks and LLMs of varying sizes. Further analysis reveals that the top-retrieved examples share similar input patterns or the same labels as the testing example. Our model is particularly effective for classification tasks with ample training examples. In contrast, tasks such as closed-book question answering and commonsense reasoning rely more on the inherent capabilities of LLMs and are less sensitive to the quality of in-context examples.

2 Related Work

In-Context Learning is an emergent property of large language models (LLMs) that enables them to perform various tasks conditioned on a few input-output examples, without any parameter updates or fine-tuning. This property has been demonstrated in LLMs such as GPT-3 (Brown et al., 2020), GPT-Neo (Black et al., 2021), and LLaMA (Touvron et al., 2023), and attracts considerable attention from the research community. One area of research is focused on understanding the underlying mechanism and principles of in-context learning. For instance, Xie et al. view in-context learning as implicit Bayesian inference, while Dai et al. interpret it as meta optimization.

Another area of research is to explore different strategies for selecting and designing in-context examples for LLMs. Recent studies (Liu et al., 2022; Rubin et al., 2022; Li et al., 2023; Luo et al., 2023) have shown that using BM25 algorithm or fine-tuning dense retrievers based on LLM feedback to retrieve from the training set can improve the performance of in-context learning. Our work also falls into this area by proposing a novel training method. To model the interaction between in-context examples, determinantal point process (Ye et al., 2023) and sequential decision-making (Zhang et al., 2022) are introduced as preliminary explorations. In contrast, Structured Prompting (Hao et al., 2022) breaks the limitation of input context length and scales the number of in-context examples to thousands.

Dense Retrieval is a widely used information retrieval approach that utilizes dense vectors to perform semantic matching between queries and documents in the latent space (Reimers and Gurevych, 2019; Wang et al., 2022). Compared to sparse retrieval methods such as BM25, dense retrieval exploits the powerful modeling capacity of pre-trained language models (PLMs) (Devlin et al., 2019) to learn relevance functions and has the potential to overcome the vocabulary mismatch problem. Various techniques such as hard negative mining (Karpukhin et al., 2020), knowledge distillation (Ren et al., 2021), and continual pre-training (Wang et al., 2022) have been proposed to enhance the performance of dense retrieval.

Retrieval Augmented LLMs combine the generative power of LLMs with the ability to retrieve relevant information from external sources (Ram et al., 2023; Lewis et al., 2020; Shi et al., 2023). This paradigm has the potential to enhance the factual consistency of generated texts, make LLMs aware of the up-to-date knowledge, as well as provide a natural way for source attribution (Nakano et al., 2021). The retrieved information can be incorporated into LLMs through various mechanisms, such as input concatenation (Shi et al., 2023), intermediate attention fusion (Borgeaud et al., 2022), and output interpolation (Khandelwal et al., 2020). For in-context learning, the goal of retrieval augmentation is to improve the performance of LLMs on downstream tasks by retrieving informative examples (Li et al., 2023; Luo et al., 2023).

3 Preliminaries

In this section, we provide a brief introduction to the problem setting of in-context example retrieval. Given a test example x_{test} from a target task and k in-context examples $\{(x_i, y_i)\}_{i=1}^{k}$ from a pre-defined pool $\mathbb{P}$, a frozen language model M is employed to predict an output y'_{test} through autoregressive decoding. The primary objective of in-context example retrieval is to retrieve k examples from $\mathbb{P}$ such that the predicted output y'_{test} is as close as possible to the ground-truth output y_{test} based on some task-specific metrics. In this paper, the example pool $\mathbb{P}$ is the union of the training set for all the tasks in our evaluation.

Straightforward solutions include utilizing the BM25 algorithm or readily available text embedding models (Wang et al., 2022; Liu et al., 2022) to retrieve examples from $\mathbb{P}$ by treating x_{test} as a query. Despite their simplicity, these methods have been shown to be more effective empirically when compared to the random selection baseline. In contrast, our framework aims to learn a dense retriever customized for in-context example retrieval by leveraging the feedback from LLMs.

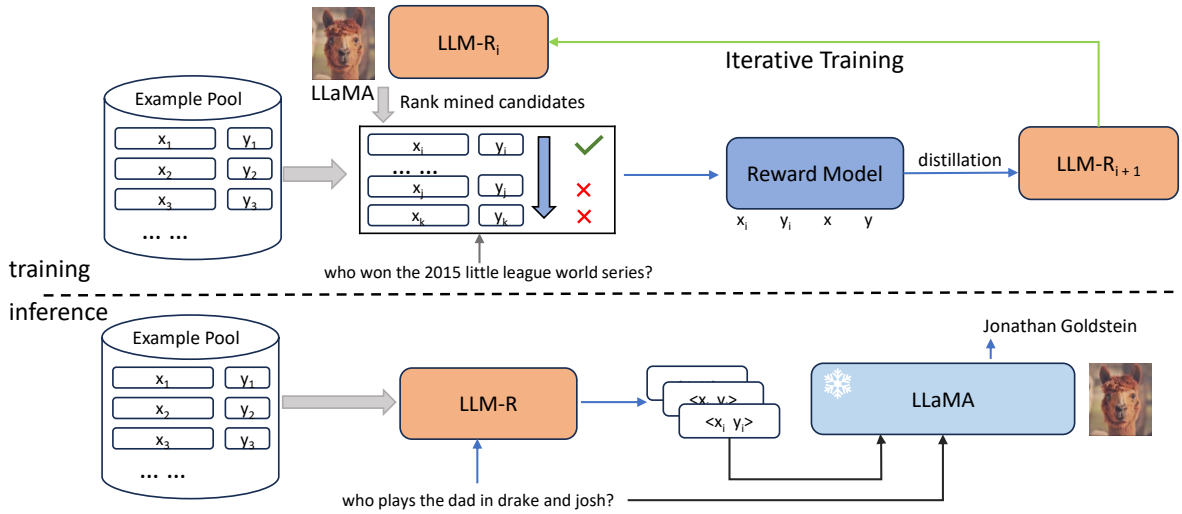

Figure 1: The overall architecture of our proposed framework LLM-R. The training process comprises three stages: generating training data based on an initial retriever and LLM feedback, reward modeling, and training dense retrievers by distilling the knowledge from the reward model. At inference time, the trained dense retriever is employed to retrieve in-context examples from the pool $\mathbb{P}$ and the retrieved examples are fed to the LLM to generate the output.

4 Methodology

Our proposed framework is depicted in Figure 1. It includes four main components: training data generation, reward modeling, dense retriever training, and inference, which are described in detail in the following subsections.

4.1 Training Data Generation

Initial Candidates Retrieval Given an example (x, y) from the training set, where x is the input and y is the groundtruth output, we retrieve the top-n candidates $\{(x_i, y_i)\}_{i=1}^{n}$ from the example pool $\mathbb{P}$ using an initial retriever. The pool $\mathbb{P}$ contains the training examples from a mixture of tasks. Since $(x, y) \in \mathbb{P}$ holds during training, we exclude itself from the retrieval results.

In this paper, we employ the unsupervised BM25 algorithm as the initial retriever. The query only consists of the input x, while each retrieval candidate is the string concatenation of the input x_i and the output y_i. This setting aligns with the test-time scenario, where the groundtruth output is unavailable. Assuming the initial retriever is reasonably effective, we anticipate that the top-n candidates would contain some positive examples and hard negative examples.

Ranking Candidates using LLMs To assess the quality of the retrieved candidates, we utilize feedback signals from a frozen LLM. Specifically, we rank the candidates in descending order based on the log-likelihood of the groundtruth output y, as given by the following equation:

$$\log p(y|x, x_i, y_i), \forall i \in \{1, 2, \ldots, n\} \quad (1)$$

Here, $p(y|x, x_i, y_i)$ is the conditional probability of y given the input x and the i-th candidate (x_i, y_i). It is noteworthy that computing $p(y|x, x_i, y_i)$ requires only one forward pass, and does not rely on any task-specific metrics, despite the autoregressive nature of language models. In practical applications, this helps reduce the inference cost of LLMs.

4.2 Reward Modeling

In order to capture the preferences of LLMs over the retrieved candidates and provide fine-grained supervision for dense retrievers, we propose to train a cross-encoder based reward model. For a training example (x, y), we first sample one positive example (x^+, y^+) from the top-ranked candidates and N_{neg} hard negative examples $\{(x_i^-, y_i^-)\}_{i=1}^{N_{\text{neg}}}$ from the bottom-ranked candidates. The reward model takes as input the concatenation of (x, y, x^+, y^+) and produces a real-valued score $s(x, y, x^+, y^+)$, similarly for the hard negatives. It is trained to

minimize the following cross-entropy loss:

$$\mathcal{L}_{\text{reward}} = -\log \frac{e^{s(x,y,x^+,y^+)}}{e^{s(x,y,x^+,y^+)} + \sum_{i=1}^{N_{\text{neg}}} e^{s(x,y,x_i^-,y_i^-)}} \quad (2)$$

It is important to note that the reward model is only used to provide supervision for the dense retriever and has access to the groundtruth label y, which is not available at test time. This is a key difference from the re-ranker in the ad-hoc retrieval setting (Ren et al., 2021). Compared to the bi-encoder based dense retrievers, the reward model enables full interaction between the inputs and can therefore serve as a teacher model.

4.3 Training LLM Retrievers with Knowledge Distillation

To facilitate efficient inference, the dense retriever is based on the bi-encoder architecture. Given a query x, we compute its low-dimensional embedding $\mathbf{h}_x$ by performing average pooling over the last-layer hidden states. Similarly, we obtain the embedding $\mathbf{h}_{(x_i,y_i)}$ for the candidate (x_i, y_i) by taking the concatenation of x_i and y_i as input. The matching score $f(x, x_i, y_i)$ is computed as the temperature-scaled cosine similarity $\cos(\mathbf{h}_x, \mathbf{h}_{(x_i,y_i)})/\tau$, where τ is a temperature hyperparameter. In this paper, we use a shared encoder for both the query and the retrieval candidates.

The dense retriever is trained to distill the knowledge from the reward model. We use the KL divergence loss $\mathcal{L}_{\text{distill}} = \text{KL}(p_{\text{reward}} \,||\, p_{\text{retriever}})$ to measure the mismatch between the reward model distribution p_{reward} and the retriever distribution $p_{\text{retriever}}$. $\mathcal{L}_{\text{distill}}$ is only computed over the hard negatives for efficiency reasons. To incorporate the in-batch negatives, we also include an InfoNCE-based contrastive loss $\mathcal{L}_{\text{cont}}$ (Chen et al., 2020) by treating the candidate with the highest reward as the positive example. The final loss function $\mathcal{L}_{\text{retriever}}$ is a weighted sum of the contrastive loss and the knowledge distillation loss:

$$\mathcal{L}_{\text{retriever}} = \alpha \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{distill}} \quad (3)$$

Here, α is a constant that controls the relative importance of the two losses.

Iterative Training As illustrated in Figure 1, the retriever trained in iteration i can be employed to retrieve candidates for the subsequent iteration $i+1$. In the first iteration, the candidates are retrieved using BM25. Such an iterative training approach (Xiong et al., 2021; Li et al., 2023) allows improving retriever quality by mining better positive and hard negative examples.

4.4 Evaluation of LLM Retrievers

Given a test example x_{test}, we compute its embedding $\mathbf{h}_{\text{test}}$ using the trained retriever and retrieve the top k candidates from the pool $\mathbb{P}$ as the k-shot in-context examples. The input to the LLM is the concatenation of the k-shot examples and x_{test}. The overall procedure is illustrated in Figure 1.

Depending on the task type of x_{test}, different decoding strategies are employed to generate the final prediction. For classification tasks, we use greedy search with constrained decoding to make sure the prediction is a valid class label. For multiple choice tasks, all the choices are ranked based on the average token-level log-likelihood score, and the one with the highest score is selected as the model's prediction. Generation tasks use greedy search without any constraints. For quantitative evaluation, the prediction is compared with the groundtruth y_{test} using task-specific metrics.

5 Experiments

5.1 Evaluation Setup

We utilize a total of 30 publicly available datasets [1] from 9 distinct categories for training and evaluation, as shown in Figure 2. This collection is based on FLAN (Wei et al., 2022) and UPRISE (Cheng et al., 2023). Different from our work, FLAN is focused on fine-tuning language models to follow instructions, while UPRISE is designed for cross-task retrieval. To test the generalization ability of the models to unseen tasks, we held out four datasets, namely QNLI, PIQA, WSC273, and Yelp, from the training process. The retrieval pool is created by taking the union of all the training examples, which results in a total of approximately 6.3M examples. For each dataset, we sample a maximum of 30k examples for training and 10k examples for evaluation to reduce the cost of LLM inference. For evaluation, we report the average metrics in each task category. Please check Table 8 for the specific metrics used for each dataset.

In the main experiments, we use LLaMA-7B (Touvron et al., 2023) as the default LLM for candidate ranking and task evaluation unless other-

[1] We use "datasets" and "tasks" interchangeably.

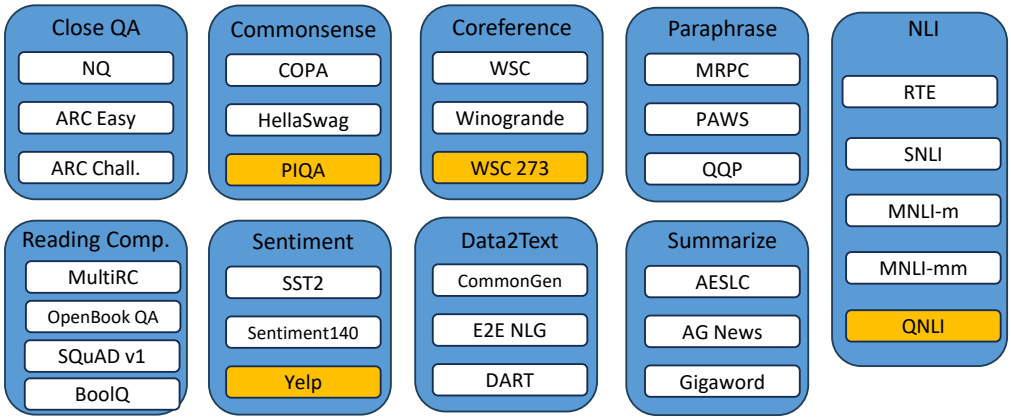

Figure 2: The collection of datasets used in our experiments. The yellow-colored datasets are held out and excluded from training. For further information, please refer to Table 8 in the Appendix.

# of datasets →	CQA 3	Comm. 3	Coref. 3	NLI 5	Para. 3	RC 4	Sent. 3	D2T 3	Summ. 3	Avg 30
Zero-shot	29.0	71.5	66.8	44.0	60.0	41.3	50.5	25.6	17.5	44.9
Random	40.4	77.6	67.2	50.9	56.6	58.1	88.8	47.0	38.9	57.9
K-means	41.6	79.5	66.0	50.8	52.6	53.6	90.9	42.5	40.5	57.0
BM25	45.9	78.1	62.9	54.7	66.1	59.9	89.6	49.3	50.0	61.3
$E5_{base}$	**49.0**	79.8	64.6	53.6	58.0	60.2	**94.4**	48.0	50.0	61.4
SBERT	48.5	79.3	64.2	57.5	64.1	**60.6**	91.9	47.4	49.3	62.1
EPR†	48.4	79.3	64.4	64.3	65.1	59.8	91.7	49.7	50.0	63.5
LLM-R (1 iter)	48.8	80.1	67.6	71.9	66.5	60.0	93.5	**50.1**	50.8	65.7
LLM-R (2 iter)	48.7	**80.4**	70.4	72.5	71.5	59.0	93.6	49.9	**51.1**	**66.5**
LLM-R (3 iter)	48.9	80.0	**70.8**	**72.6**	**72.8**	58.0	92.9	49.8	50.8	66.4
Std dev.	±0.2	±0.8	±0.7	±0.1	±1.1	±0.0	±0.4	±0.0	±0.1	±0.2

Table 1: Our main results. We report the average metrics for Close QA (CQA), Commonsense Reasoning (Comm.), Coreference (Coref.), NLI, Paraphrase (Para.), Reading Comprehension (RC), Sentiment (Sent.), Data-to-text (D2T), Summarize (Summ.). The standard deviation is computed over 3 runs with the "Random" baseline. Dense retriever baselines include E5 (Wang et al., 2022), SBERT (Reimers and Gurevych, 2019), and EPR (Rubin et al., 2022). †: Our re-implementation for fair comparison.

wise specified. The reward model is initialized with $ELECTRA_{base}$ (Clark et al., 2020) and the retriever is initialized with $E5_{base}$ (Wang et al., 2022). The baselines include zero-shot prompting, k-means clustering, random selection, BM25 (Lin et al., 2021), and two off-the-shelf dense retrievers, namely SBERT (all-mpnet-base-v2) (Reimers and Gurevych, 2019) and $E5_{base}$. Except for zero-shot evaluation, we retrieve 8 in-context examples for each test input. More implementation details and training hyperparameters can be found in Appendix A.

5.2 Main Results

Table 1 presents the main results of our experiments. We observe that the simple BM25 algorithm serves as a strong baseline, exhibiting consistent improvements over the random selection strategy. This conclusion aligns with the findings of Luo et al.. After the first iteration, our proposed model LLM-R outperforms all the baselines (63.5 → 65.7) by training on the BM25 retrieved candidates. The second iteration includes the mined positive and hard negative examples from "LLM-R (1 iter)", raising the average score to 66.5 (+0.8). Further iterations do not yield substantial improvements, indicating that the model has converged.

6 Analysis

In this section, we examine the performance of LLM-R across various tasks, LLMs, and model variants. Unless explicitly specified, "LLM-R" refers to the model with 2 training iterations.

	CQA	Comm.	Coref.	NLI	Para.	RC	Sent.	D2T	Summ.	Avg
LLM-R (1 iter)	**48.8**	**80.1**	67.6	**71.9**	66.5	60.0	**93.5**	**50.1**	**50.8**	**65.7**
model variants										
w/o reward model	**48.8**	79.1	64.3	68.9	70.2	60.5	91.7	49.4	50.5	64.9
LLM score as reward	48.0	79.4	67.0	67.0	**74.0**	60.5	91.5	49.6	50.3	65.2
retriever initialization										
initialize w/ $\text{BERT}_{\text{base}}$	48.7	79.6	**69.4**	70.9	63.0	**60.7**	92.0	50.0	50.2	65.2

Table 2: Different training variants of LLM-R. "w/o reward model" is trained solely with contrastive loss on LLM ranked candidates. "LLM score as reward" uses the log-likelihood score from LLMs as the distillation target. Neither of these variants utilizes the reward model.

	Zero-shot	Random	K-means	BM25	E5_{base}	SBERT	LLM-R
QNLI	49.2	56.4	53.4	62.2	61.5	61.9	**69.6**$^{\uparrow 7.7}$
PIQA	77.0	79.1	79.4	81.3	81.3	80.7	**81.6**$^{\uparrow 0.3}$
WSC273	74.0	74.4	74.7	64.5	65.2	62.6	**79.5**$^{\uparrow 4.8}$
Yelp	47.9	92.0	93.5	93.5	**97.3**	95.9	95.9$^{\downarrow 1.4}$
Average	62.0	75.5	75.3	75.4	76.3	75.3	**81.7**$^{\uparrow 5.4}$

Table 3: Generalization to four held-out tasks.

6.1 Training Pipeline of LLM-R

We investigate several LLM-R variants LLM-R in Table 2 to understand the contribution of each component. The "w/o reward model" variant removes the knowledge distillation loss and sees 0.8 points drop in average score. This indicates that the reward model is crucial for the performance of LLM-R. Inspired by REPLUG (Shi et al., 2023), we experiment with a variant that uses the log-likelihood from LLMs as the reward for distillation. Although it outperforms the "w/o reward model" variant, it still lags behind our method by 0.5 points. We hypothesize that the log-likelihood of LLMs may not be well-calibrated for knowledge distillation with KL divergence. Changing the retriever initialization from E5 (Wang et al., 2022) to BERT (Devlin et al., 2019) results in a performance drop, but not as significant as in the ad-hoc retrieval setting.

6.2 Generalization Ability of LLM-R

We evaluate the generalization ability of LLM-R from two dimensions. In the first scenario, we test whether the trained retriever can retrieve good in-context examples for tasks that are not seen during training. In the second scenario, we test whether a model trained with one LLM can generalize to other LLMs that vary in size and quality.

In Table 3, we report the performance of LLM-R on four held-out tasks. The results demonstrate that LLM-R surpasses the second-best model E5_{base} by an average of 5.4 points, indicating its ability to generalize to previously unseen tasks. Under the current evaluation protocol, there are training datasets that share the same task category as the held-out ones (e.g., QNLI and SNLI are both for natural language inference). A more challenging setting is to test on non-overlapping task categories, which we leave for future work.

The LLM-R model is trained with LLaMA-7B. To evaluate its generalization ability across different LLMs, we test on three other models, namely GPT-Neo-2.7B (Black et al., 2021), LLaMA-13B, and GPT-35-Turbo. Results in Table 4 show that LLM-R consistently outperforms the BM25 baseline for LLMs with parameter ranges from 2.7B to tens of billions. Notably, the gains are particularly significant for small-size language models, possibly because they are less powerful and thus require higher-quality examples to perform in-context learning.

6.3 When does LLM-R Work and When Does it Not?

Reporting a single aggregate score for all tasks facilitates comparison across different model variants. However, this approach hides the fact that LLM-R performs better on certain tasks than others, and may even lead to performance degradation in some cases. In Figure 3, we partition the tasks into two groups. A task is considered to be *knowledge-intensive* if solving this task requires commonsense, complex reasoning, or memorized factual knowledge.

For tasks in the knowledge-intensive set, the

	CQA	Comm.	Coref.	NLI	Para.	RC	Sent.	D2T	Summ.	Avg
gpt-neo-2.7b										
BM25	41.1	67.0	53.2	47.6	64.5	51.2	78.3	45.4	47.3	54.4
LLM-R	42.2	68.0	59.7	71.5	73.0	51.6	91.6	46.9	48.8	**61.8**$^{\uparrow 7.4}$
llama-13b										
BM25	49.6	80.1	61.1	67.0	69.9	60.5	92.5	49.9	50.9	64.6
LLM-R	52.0	83.7	71.2	76.8	73.3	62.2	94.2	50.7	52.0	**68.8**$^{\uparrow 4.2}$
gpt-35-turbo†										
BM25	75.3	85.2	65.0	78.1	78.0	84.4	95.7	51.9	52.8	74.7
LLM-R	79.3	86.7	63.8	79.6	76.0	84.0	95.4	52.2	53.0	**75.1**$^{\uparrow 0.4}$

Table 4: Generalization to LLMs that are not used for training. †: Since the official API of *gpt-35-turbo* does not return the log-probabilities, we use different input-output templates to formulate all tasks as text generation. Consequently, the scores of *gpt-35-turbo* cannot be directly compared with those of other LLMs. More details are in Appendix B.

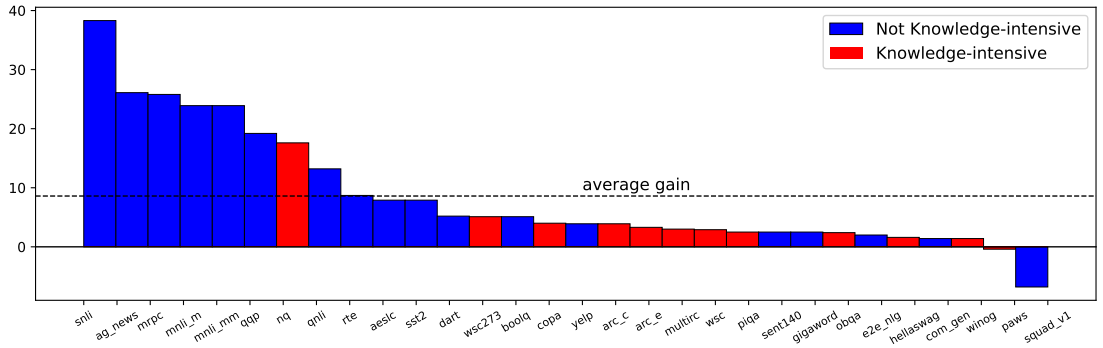

Figure 3: Performance gains of LLM-R over the random selection baseline. The selected *knowledge-intensive* tasks are NQ, ARC (easy and challenge), PIQA, HellaSwag, COPA, Paws, OpenBook QA, WSC273, WSC, Winogrande, and MultiRC.

absolute improvements are substantially smaller than the average, with NQ being the only exception. This is not surprising, as these tasks rely more heavily on the underlying foundation model's capability to perform reasoning and knowledge memorization. For the NQ dataset, we empirically find that there is some overlap between the training and test sets, where test questions are paraphrases of some training questions. Despite this, we decide to keep the NQ dataset in our evaluation, as it is a widely used benchmark and the remaining non-overlapping questions are still valuable.

Another noticeable case is the SQuAD v1 dataset (Rajpurkar et al., 2016), where LLM-R performs worse than the random selection baseline. Upon manual inspection, we find that many questions in SQuAD share the same passage as the context. This frequently results in LLM-R retrieving examples with limited diversity, which may account for the observed decline in performance.

In Table 5, for the Sentiment140 and MNLI datasets, our model helps by retrieving examples that share similar input patterns with the test example. In contrast, the PIQA dataset requires commonsense knowledge and may not benefit much from the retrieved examples.

6.4 Using Different LLMs for Data Generation and Task Evaluation

One crucial aspect of our framework is the selection of the LLM for training data generation and task evaluation. During the training phase, the LLM plays a pivotal role in ranking the retrieved candidates and providing supervision signals for the reward model. In the task evaluation phase, the LLM is used to generate the final predictions.

We experiment with GPT-Neo-2.7B and LLaMA-7B. Table 6 shows the results under different combinations of LLMs for training and evaluation. We observe that the quality of the evaluation LLM is the primary determinant for the final performance, while the choice of ranking LLM has a relatively minor impact. Although merging the training data from two LLMs yields

Task name	Sentiment140
Test Input	Math review. Im going to fail the exam. What is the sentiment of this tweet?
Test Answer	**Negative**
LLM-R	revising for maths exam on tuesday which im gonna fail badly What is the sentiment of this tweet? **Negative**
Task name	MNLI-m
Test Input	"Part 2), Confidentiality of Alcohol and Drug Abuse Patient Records." Hypothesis: "Drug and alcohol patient records should be confidential" Does the premise entail the hypothesis? Yes, No, or Maybe?
Test Answer	**Yes**
LLM-R	Premise: "Eligible Clients unable to attain needed legal assistance" Hypothesis: "Clients that should have received legal assistance but didn't" Does the premise entail the hypothesis? Yes, No, or Maybe? **Yes**
Task name	PIQA
Test Input	Here is a goal: "How can I keep a bathroom mirror from fogging up?" How would you accomplish this goal?
Test Answer	**Wipe down with shaving cream.**
LLM-R	Here is a goal: "how do you 'clean up' an eyebrow you've filled in?" How would you accomplish this goal? **use concealer to cover up any mistakes made.**

Table 5: Retrieved examples by LLM-R. The bold texts are the groundtruth answers for the test inputs and retrieved candidates. More examples are available in Table 11.

Rank LLM → Eval LLM ↓	GPT-Neo-2.7B	LLaMA-7B	Both
GPT-Neo-2.7B	**61.7**	61.3	61.6
LLaMA-7B	66.0	65.7	**66.3**

Table 6: On the impacts of using different LLMs for candidate ranking and task evaluation. The "Both" setting merges the training data from two LLMs.

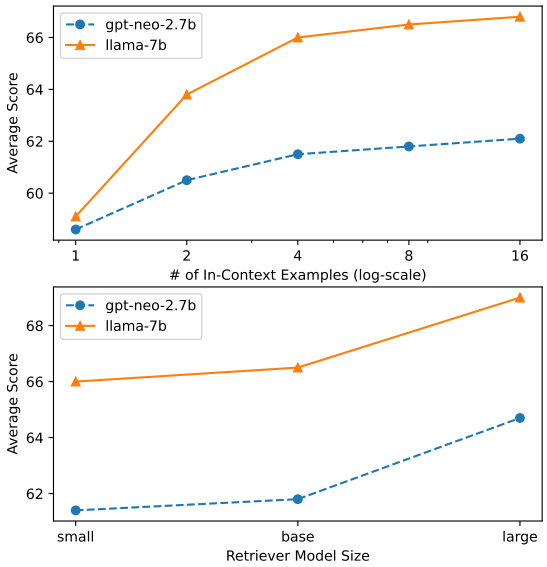

Figure 4: The scaling effect with respect to the number of in-context examples and retriever size. Our main experiments use 8 in-context examples and base-size retriever. We vary the retriever model size by initializing with the released E5-{small, base, large} checkpoints from Wang et al..

the best overall performance, we do not employ this technique in our main experiments for the sake of simplicity.

6.5 Scaling the Number of In-Context Examples and Retriever Size

In Figure 4, we investigate the scaling effect of LLM-R from two aspects: the number of in-context examples and the retriever model size. The overall performance improves as we increase the number of retrieved examples, but the gains diminish after 4 examples. Including more examples usually leads to longer prompts and higher inference cost.

With regard to the retriever size, we observe that the small-size model produces comparable results with the base-size one, whereas the large-size retriever exhibits a more substantial performance boost. The trends are consistent for the two examined language models. Practitioners can select the appropriate configurations based on the trade-off between performance and computational cost.

7 Conclusion

In this paper, we introduce an iterative training framework named *LLM-R* to retrieve high-quality in-context examples for large language models. This framework generates training data by utilizing a frozen LLM to rank the top retrieved candidates, and then learns a cross-encoder based reward model to capture the ranking preference. Bi-encoder based dense retrievers are trained to distill the knowledge from the reward model. We conduct a comprehensive evaluation of LLM-R on a diverse set of tasks and demonstrate that it consistently outperforms various strong baselines. Our model also generalizes well to held-out tasks and LLMs of varying sizes.

Limitations

In our framework, we treat each candidate example independently and retrieve the top-k results for each test example. This may be suboptimal as the in-context examples can influence each other. Incorporating the techniques from the field of combinatorial optimization can be a promising direction to explore.

Another limitation of our study is related to the automatic evaluation protocol. To compare the performance of different methods, we report the arithmetic mean of the metrics over all tasks. However, this may put generation tasks at a disadvantage since metrics like ROUGE and BLEU typically have a narrower range of variation compared to classification accuracy. Moreover, the simple arithmetic mean does not account for the quality of each dataset.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer ai2 reasoning challenge. *ArXiv preprint*, abs/2102.03315.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George 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 W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yu-Wei Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Denvy Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. *ArXiv preprint*, abs/2303.08518.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-

context? language models secretly perform gradient descent as meta optimizers. *ArXiv preprint*, abs/2212.10559.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *ArXiv preprint*, abs/2212.06713.

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.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Patrick S. H. 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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. *ArXiv preprint*, abs/2305.04320.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Jimmy J. Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, Rodrigo Nogueira, and David R. Cheriton. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *ArXiv preprint*, abs/2305.14128.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv preprint*, abs/2112.09332.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

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*, abs/2301.12652.

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 *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv preprint*, abs/2212.03533.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. *ArXiv preprint*, abs/2302.05698.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 446–456, Florence, Italy. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Implementation Details

	Retriever	Reward Model
initialization	$E5_{base}$	$ELECTRA_{base}$
learning rate	3×10^{-5}	10^{-5}
# of GPUs	8	8
batch size	256	128
train steps	6k	3k
τ	0.01	n.a.
α	0.2	n.a.
positive examples	top 3	bottom 16
negative examples	top 3	bottom 16
# of negatives	3	7
ranking depth	100	100
input length	256	384

Table 7: Hyperparameters for training the bi-encoder retriever and reward model. We use the same hyperparameters for every iteration.

The hyperparameters for the retriever model and reward model are summarized in Table 7. The $E5_{base}$ checkpoint is available at `https://huggingface.co/intfloat/e5-base-v2`. This checkpoint is also employed for the k-means clustering baseline, where we select 8 examples closest to each cluster center as the in-context examples. For each iteration, we employ LLaMA-7B to rank the top-100 retrieved candidates. As we retrieve from a unified pool of examples, it is possible that a candidate comes from a different task than the query. In this case, we assign a low score to it.

During the evaluation, we retrieve top-8 candidates and use them as in-context examples. The maximum input length for LLaMA-7B is set to 1024. Longer inputs are truncated from the left side. The maximum output length is set to 64. The most time-consuming part in our pipeline is ranking candidates with LLaMA-7B, which takes about 12 hours for 200k examples with 8 V100

Dataset name	Category	# train	# test	Metric	Held-out?
AESLC (Zhang and Tetreault, 2019)	Summarize	13,181	1,750	ROUGE-L	N
AGNews (Zhang et al., 2015)	Summarize	120,000	7,600	Accuracy	N
ARC Challenge (Bhakthavatsalam et al., 2021)	Close QA	1,117	1,165	Accuracy	N
ARC Easy (Bhakthavatsalam et al., 2021)	Close QA	2,241	2,365	Accuracy	N
BoolQ (Clark et al., 2019)	Reading Comp.	9,427	3,270	Accuracy	N
CommonGen (Lin et al., 2020)	Data-to-text	67,389	4,018	ROUGE-L	N
COPA (Roemmele et al., 2011)	Commonsense	400	100	Accuracy	N
DART (Nan et al., 2021)	Data-to-text	62,659	2,768	ROUGE-L	N
E2E NLG (Dušek et al., 2019)	Data-to-text	33,525	1,847	ROUGE-L	N
Gigaword (Napoles et al., 2012)	Summarize	2,044,465	730	ROUGE-L	N
HellaSwag (Zellers et al., 2019)	Commonsense	39,905	10,042	Accuracy	N
MNLI (m) (Williams et al., 2018)	NLI	392,702	9,815	Accuracy	N
MNLI (mm) (Williams et al., 2018)	NLI	392,702	9,832	Accuracy	N
MRPC (Dolan and Brockett, 2005)	Paraphrase	3,668	408	Accuracy	N
MultiRC (Khashabi et al., 2018)	Reading Comp.	27,243	4,848	F1	N
NQ (Kwiatkowski et al., 2019)	Close QA	87,925	3,610	Exact Match	N
OpenBook QA (Mihaylov et al., 2018)	Reading Comp.	4,957	500	Accuracy	N
PAWS (Zhang et al., 2019)	Paraphrase	49,401	8,000	Accuracy	N
PIQA (Bisk et al., 2020)	Commonsense	16,113	1,838	Accuracy	Y
QNLI (Rajpurkar et al., 2018)	NLI	104,743	5,463	Accuracy	Y
QQP (Wang et al., 2019)	Paraphrase	363,846	40,430	Accuracy	N
RTE (Bentivogli et al.)	NLI	2,490	277	Accuracy	N
Sentiment140 (Go et al., 2009)	Sentiment	1,600,000	359	Accuracy	N
SNLI (Bowman et al., 2015)	NLI	549,367	9,824	Accuracy	N
SQuAD v1 (Rajpurkar et al., 2016)	Reading Comp.	87,599	10,570	Exact Match	N
SST2 (Socher et al., 2013)	Sentiment	67,349	872	Accuracy	N
Winogrande (Sakaguchi et al., 2020)	Coreference	40,398	1,267	Accuracy	N
WSC (Levesque et al., 2012)	Coreference	554	104	Accuracy	N
WSC273 (Levesque et al., 2012)	Coreference	0	273	Accuracy	Y
Yelp (Zhang et al., 2015)	Sentiment	490,456	33,285	Accuracy	Y
Total	n.a.	6.3M	177k	n.a.	n.a.
Total (sampled)	n.a.	591k	123k	n.a.	n.a.

Table 8: Statistics for the datasets used in this paper.

GPUs. Training the retriever model and reward model takes less than 10 hours in total.

B Evaluation with GPT-35-Turbo

Due to quota limits, we sample at most 1k examples for each dataset. As GPT-35-Turbo does not return token-level log-probabilities, we cannot evaluate the multiple-choice datasets by computing the log-likelihood of each option. Instead, we append all the options to the end of the input, and let the model generate the option index. An example is shown in Table 9. We also tried using this format to LLaMA-7B, but the performance is significantly worse than comparing the log-likelihood of each option.

For a small number of test examples, GPT-35-Turbo fails to follow the patterns of in-context examples and generates outputs that are not valid class labels. We add some simple heuristics based on string matching to determine the model prediction.

Input	What happens next in this paragraph? How to survive remedial classes Look at the course as an opportunity. Many students are discouraged when they are assigned to a remedial class. Some assume this placement means they aren't ready for college. OPTIONS: A) However, people who are not unable to do what they're given on campus, or those who are cut out from college academies, are likely to have some little snitches. You want to be prepared for a negative outcome if possible. B) In this case, you should consider what you will do if your subject consists of a certain term or number of subject areas. You could set up a study study program yourself or tutor a student who is struggling to thoroughly comprehend where they sat for homework. C) If you take the course, you might find you feel highly motivated after passing the test. Try to develop a positive attitude towards the course so that you are not discouraged when you take your homework at the end of the day. D) However, being assigned a remedial class doesn't mean that you are behind, just that you have an opportunity to receive better instruction and improve your skills in a subject that you have struggled with in the past. There is nothing unusual about being asked to attend a remedial course: two thirds of community college students take at least one remedial course.
Output	D

Table 9: Input-output format for GPT-35-Turbo. This example is from the HellaSwag dataset. We add some line breaks for better readability.

Task	Zero-shot	Random	Kmeans	BM25	$E5_{base}$	SBERT	EPR	LLM-R		
								1 iter	2 iter	3 iter
AESLC	5.8	19.4	19.0	26.8	27.0	25.3	26.0	26.7	27.3	27.1
AGNews	31.5	67.4	71.9	90.6	90.6	90.2	91.8	92.4	93.5	93.5
ARC Chall.	35.6	39.7	40.5	40.3	44.6	42.8	43.0	43.4	43.6	44.0
ARC Easy	51.0	60.0	61.8	59.9	63.0	63.1	63.1	63.6	63.3	63.6
BoolQ	64.7	70.0	69.0	74.7	72.4	73.9	74.8	75.6	75.1	74.1
CommonGen	19.2	36.3	34.4	37.6	37.4	37.6	39.2	38.2	37.7	37.3
COPA	66.0	80.0	85.0	78.0	83.0	82.0	82.0	84.0	84.0	84.0
DART	22.9	52.0	46.6	55.9	54.7	54.4	56.2	57.3	57.2	57.3
E2E NLG	34.6	52.7	46.4	54.5	51.8	50.2	53.6	54.9	54.7	54.9
Gigaword	15.3	30.0	30.7	32.7	32.5	32.6	32.4	33.3	32.5	31.8
HellaSwag	71.5	73.9	74.0	74.9	75.2	75.3	75.2	75.4	75.5	75.4
MNLI (m)	35.8	46.3	44.2	50.1	44.5	50.8	59.9	68.2	70.2	69.8
MNLI (mm)	35.6	48.1	45.4	48.3	44.7	49.3	61.5	69.5	72.0	71.3
MRPC	69.1	49.5	38.0	61.8	41.2	52.7	55.9	62.3	75.3	78.2
MultiRC	57.0	48.5	34.1	54.2	56.0	55.3	50.4	52.9	51.5	52.1
NQ	0.3	21.5	22.6	37.6	39.3	39.4	39.2	39.4	39.1	39.2
OpenBook QA	41.6	49.8	49.0	49.6	51.4	51.4	49.6	50.8	52.2	53.4
PAWS	53.2	57.0	56.6	56.6	55.4	58.2	57.7	57.0	56.6	57.0
PIQA	77.0	79.1	79.4	81.3	81.3	80.7	80.5	80.9	81.6	80.6
QNLI	49.2	56.4	53.4	62.2	61.5	61.9	65.0	74.4	69.6	69.4
QQP	57.7	63.4	63.3	79.8	77.5	81.3	81.7	80.1	82.6	83.3
RTE	59.6	59.9	58.5	65.7	63.9	67.2	66.8	67.2	68.6	70.4
Sentiment140	49.3	88.6	89.4	90.8	93.9	92.2	91.4	90.8	91.1	90.3
SNLI	39.8	43.7	52.5	47.1	53.5	58.4	68.4	80.2	82.0	82.2
SQuAD v1	2.1	64.1	62.3	61.2	60.8	61.6	64.3	60.7	57.3	52.5
SST2	54.4	85.9	89.7	84.4	92.1	87.6	88.7	94.0	93.8	93.1
Winogrande	62.0	66.7	66.5	67.5	66.9	66.5	66.5	67.9	68.1	67.2
WSC	64.4	60.6	56.7	56.7	61.5	63.5	61.5	60.6	63.5	66.4
WSC273	74.0	74.4	74.7	64.5	65.2	62.6	65.2	74.4	79.5	78.8
Yelp	47.9	92.0	93.5	93.5	97.3	95.9	95.1	95.7	95.9	95.5
Average	44.9	57.9	57.0	61.3	61.4	62.1	63.5	65.7	66.5	66.4

Table 10: Detailed results for each dataset.

Task Name	AG News
Test Input	"Holiday Shoppers Off to a Fast Start Holiday shoppers spent 10 percent more Friday than they did a year ago, according to early reports, but Wal-Mart Stores Inc. dampened hopes for a strong start to the key retail season by " What is this text about? World, Sports, Business, or Technology?
Test Answer	**Business**
LLM-R Top 1	"Disappointing holiday news hurts retail shares Shares in a range of area retailers dipped Monday on disappointing Thanksgiving sales data from Wal-Mart Stores Inc. In addition, ShopperTrak, which tallies sales results from 30,000 stores nationwide, said " What is this text about? World, Sports, Business, or Technology? **Business**
Task name	ARC Challenge
Test Input	In the 17th century, to estimate the distance to other planets, scientists first used the technique of viewing the planet from two different locations on Earth's surface. Which characteristic of the planet were the scientists using to calculate the distance from Earth?
Test Answer	**location**
LLM-R Top 1	Which physical characteristic of Earth is similar to a physical characteristic of the Moon? **its mountain ranges**
Task name	ARC Easy
Test Input	What is the major cause of seasonal changes?
Test Answer	**tilt of the Earth's axis**
LLM-R Top 1	Which occurs as a result of Earth's tilt on its rotating axis? **seasonal changes in the climate**
Task name	CommonGen
Test Input	Concepts: field, throw, kid, bunch, ball. Write a sentence that includes all these words.
Test Answer	**A bunch of kids are running around and throwing a ball on a field.**
LLM-R Top 1	Concepts: look, ball, lot. Write a sentence that includes all these words. **Two babies look up while they are playing in a playpen with a lot of balls.**
Task name	COPA
Test Input	"The boy skipped dinner." What is the cause?
Test Answer	**He ate a big lunch.**
LLM-R Top 1	"The parents left their children with a babysitter." What is the cause? **They made plans to celebrate their anniversary.**
Task name	DART
Test Input	Triple: The Mill, eatType, coffee shop; The Mill, food, Chinese; The Mill, priceRange, moderate; The Mill, area, city centre; The Mill, near, The Sorrento What is a sentence that describes this triple?
Test Answer	**There is a coffee shop serving Chinese food called The Mill. It has a moderate price range is is find in the city centre near The Sorrento.**
LLM-R Top 1	Triple: The Mill, eatType, coffee shop; The Mill, food, Indian; The Mill, priceRange, cheap; The Mill, area, riverside; The Mill, near, The Sorrento What is a sentence that describes this triple? **The Mill coffee shop is located in the riverside area near The Sorrento. They serve Indian food at a cheap price.**
Task name	Gigaword
Test Input	Write a short summary for this text: the dollar and major european currencies traded within narrow ranges on tuesday on the london forex market , which was waiting for the easter holiday weekend and for us employment figures to be announced on friday , traders said in late afternoon .
Test Answer	**london forex market stable as market waits for easter us data**
LLM-R Top 1	Write a short summary for this text: the dollar was stable over-all early monday afternoon by comparison with morning levels on the london forex market , which was waiting for publication at the end of the week of us inflation figures , traders said . **dollar stable in london as market waits for us inflation data**
Task name	MRPC
Test Input	Here are two sentences: An episode is declared when the ozone reaches .20 parts per million parts of air for one hour . A Stage 1 episode is declared when ozone levels reach 0.20 parts per million . Do they have the same meaning?
Test Answer	**Yes**
LLM-R Top 1	Here are two sentences: A Stage One alert is declared when ozone readings exceed 0.20 parts per million during a one-hour period . A Stage 1 episode is declared when ozone levels reach 0.20 parts per million . Do they have the same meaning? **Yes**
Task name	NQ
Test Input	Question: legislation regarding data protection and security in uk? Answer:
Test Answer	**The Data Protection Act 1998**
LLM-R Top 1	Question: which law relates to the protection of personal information? Answer: **Data Protection Act 1998**

Table 11: More retrieved examples. The format is the same as Table 5.