TDR: Task-Decoupled Retrieval with Fine-Grained LLM Feedback for In-Context Learning

Anonymous ACL submission

Abstract

In-context learning (ICL) has become a classic approach for enabling LLMs to handle various tasks based on a few input-output examples. The effectiveness of ICL heavily relies on the quality of these examples, and previous works which focused on enhancing example retrieval capabilities have achieved impressive performances. However, two challenges remain in retrieving high-quality examples: (1) Difficulty in distinguishing cross-task data distributions, (2) Difficulty in making the fine-grained connection between retriever output and feedback from LLMs. In this paper, we propose a novel framework called TDR. TDR decouples the ICL examples from different tasks, which enables the retrieval module to retrieve examples specific to the target task within a multi-task dataset. Furthermore, TDR models fine-grained feedback from LLMs to supervise and guide the training of the retrieval module, which helps to retrieve high-quality examples. We conducted extensive experiments on a suite of 30 NLP tasks, the results demonstrate that TDR consistently improved results across all datasets and achieves state-of-the-art performance. Meanwhile, our approach is a plug-and-play method, which can be easily combined with various LLMs to improve example retrieval abilities for ICL.

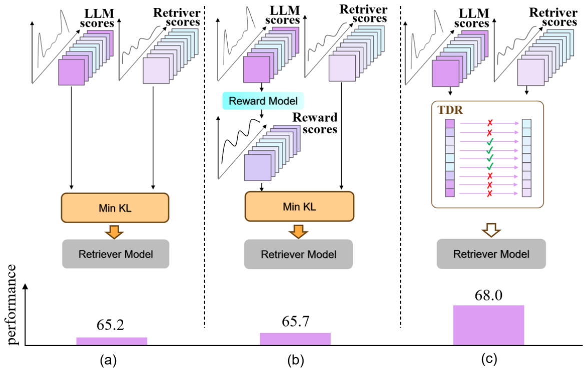

Figure 1: Comparison with previous methods. (a) KL divergence-based method: Uses LLM scores with KL divergence minimization, Performance is limited by the large distributional gap between retriever scores and LLM scores (b) Reward model-based KL method: Applies a reward model to smooth scores but still uses KL divergence, improving performance over (a) while facing similar alignment challenges. (c) Our method: Selects retrieval candidates using LLM scores, establishing positive correlation without distribution fitting, thus avoiding misalignment and improving performance.

1 Introduction

Large language models (LLMs) like GPT-4(OpenAI et al., 2024) have demonstrated exceptional performance across a wide range of language tasks. These models are typically trained on vast datasets, implicitly storing a significant amount of world or domain knowledge within their parameters. However, they are also prone to hallucinations and cannot fully represent long-tail knowledge from their training corpora(Xie et al., 2021). In-context learning (ICL)(Brown et al., 2020; Black et al., 2021; Luo et al., 2023) has emerged as a transformative approach for LLMs, enabling them to effectively leverage long-tail knowledge learned during training with minimal input-output examples, thereby significantly reducing model hallucinations without requiring any updates to model parameters. The effectiveness of ICL heavily depends on the quality of the provided examples(Liu et al., 2021; Work). As proposed by (Wang et al., 2023) and (Shi et al., 2023), the task of retrieving in-context examples for LLMs is specifically designed to improve the quality of retrieved examples. Our work builds on these foundations and focuses on enhancing the retrieval capability of high-quality in-context examples to maximize the potential and performances of LLMs.

Despite these advances, several challenges remain to understand and improve the effectiveness of ICL, which limits its potential. One such challenge is distinguishing data from different tasks. In real-world scenarios, retrieval pools often contain examples from multiple tasks, with significant differences in data distribution and characteristics. Retrieving examples from other tasks can negatively impact LLMs learning from in-context examples. However, this challenge is barely investigated in previous work. Table 7 in the Appendix shows specific examples retrieved from other tasks, which have texts similar to the query and significantly different answer patterns, making it difficult for LLMs to learn from these retrieval examples.

Another challenge is how to make the fine-grained connection between retriever output and feedback from LLMs. The relationship between the scores output by retriever and LLM feedback scores can be highly correlated. The retriever trained with LLM feedback exhibits a more consistent scoring pattern when compared to the LLM feedback scores(Wang et al., 2023). In contrast, the scatter distribution of E5(Wang et al., 2022) which is not trained with fine-grained LLM feedback shows greater fluctuation and instability. It is crucial to establish a direct and efficient relationship between the output of retriever and LLM to enhance the quality of retrieved samples.

In this paper, we propose a novel framework for retrieving high-quality in-context examples for large language models, named TDR. We start with a bi-encoder(Devlin, 2018) as the initial dense retriever to obtain a candidate set of examples. By decoupling the training of examples from different tasks, TDR enable the retriever to focus on retrieving relevant data specific to the target task within a multi-task dataset, thereby improving the precision and relevance of retrieved examples. Besides, TDR employs a specific loss function TDR to model the fine-grained feedback from LLMs and guide the training of the dense retriever. This process can be iterated multiple times to enhance the retriever's ability to retrieve high-quality examples from the specific task.

Following the task setting of (Wang et al., 2023), we conducted experiments on a dataset comprising 30 diverse NLP tasks, spanning nine categories including question answering, natural language inference, commonsense reasoning, and summarization, etc. Extensive experimental results obtained using LLaMA-7B (Touvron et al., 2023) demonstrate that our method outperforms the previous state-of-the-art approach, showing consistent improvements in in-context learning performance across all tasks. Similar gains are observed for unseen tasks during training and across LLMs of varying sizes, further validating the effectiveness and versatility of our strategy.

Contributions of this paper can be summarized as follows:

-We analyze the key factors affecting the capabilities of retrieving in-context examples for large language models and observe that distinguishing data from different tasks and making fine-grained connection between the outputs of retriever and LLMs count most.

-We propose TDR, a novel scheme to promote retrieving high-quality contextual examples for large language models. Specifically, decoupling the training of examples from different tasks is developed to further distinguishing data from different domains. Meanwhile, we employ a correlation-enhanced loss function to model the fine-grained feedback from LLMs, which can make better use of feedback from LLMs.

-Extensive evaluation on 30 NLP tasks demonstrates that TDR outperforms previous state-of-the-art method, achieving a state-of-art performance across all tasks including seen and unseen tasks during training.

2 Related Work

2.1 In-context learning

In-context learning (ICL) is an emergent capability of large language models (LLMs) that allows them to solve tasks by conditioning on input-output demonstrations without parameter updates. This phenomenon has been widely studied in models like GPT-3(Brown et al., 2020), PaLM(Chowdhery et al., 2023), and LLaMA(Touvron et al., 2023). Research on ICL primarily focuses on two directions: mechanistic interpretation and example optimization strategies.

For mechanistic understanding, Studies(Xie et al., 2021) proposes diverse theoretical frameworks and interprets ICL as implicit Bayesian inference, where models update latent task representations based on demonstrations. Concurrently, (Von Oswald et al., 2023) argues that transformers implicitly perform gradient descent during ICL, mimicking meta-optimization processes. Recent work(Park et al., 2024) further reveals that LLMs dynamically reconfigure semantic representations when contextual examples scale, shifting from pre-trained priors to task-specific structures.

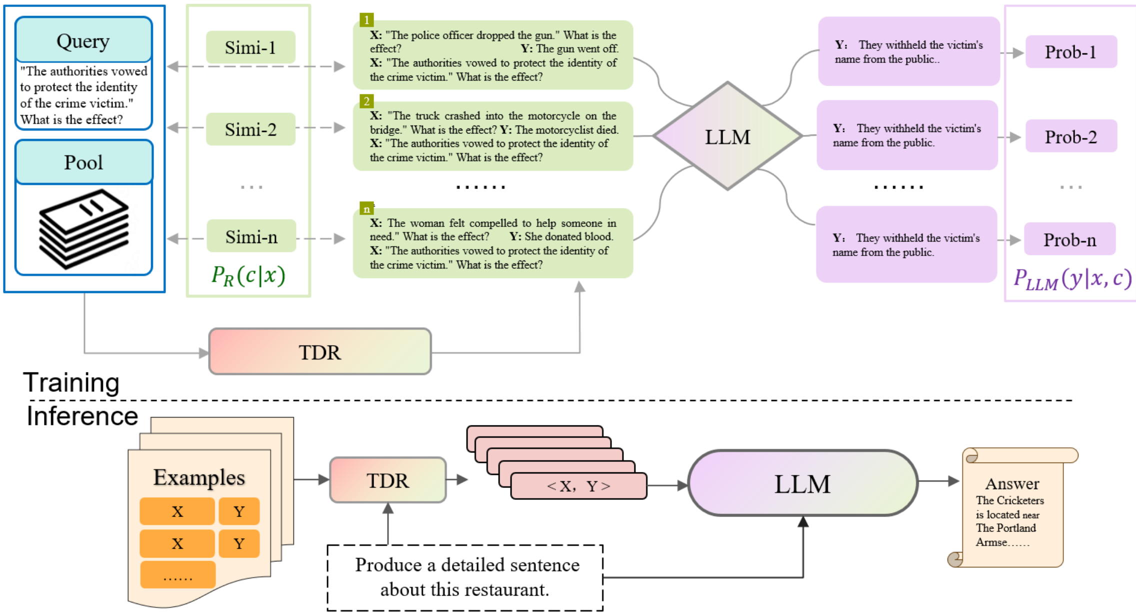

Figure 2: TDR Framework for Retriever Fine-Tuning and Inference. Training: The retriever selects task-specific examples based on queries, while the LLM generates corresponding probabilities. TDR optimizes the retriever to maximize the likelihood of correct answers given queries and examples (Section 3.3). Inference: The fine-tuned retriever retrieves in-context examples from pool $\mathbb{P}$, which are concatenated with the query and fed to the LLM for prediction.

In example optimization, researchers explore strategies to enhance ICL performance through prompt engineering and data selection. Retrieval-based methods, such as BM25-based selection(Reimers, 2019) and contrastive retrievers(Rubin et al., 2021), aim to identify semantically relevant examples. Advanced techniques like determinantal point processes(Ye et al., 2023) model inter-example interactions, while structured prompting(Hao et al., 2022) extends context length to thousands of tokens. The LLM-R framework(Wang et al., 2023) introduced a novel approach using a reward model to iteratively train dense retrievers for identifying high-quality in-context examples. Our work aligns with this direction, proposing a novel method for dynamic example selection.

2.2 Retrieval-augmented Models

Retrieval-augmented large language models (RALMs) integrate generative capabilities with external knowledge to enhance factual accuracy and timeliness(Guu et al., 2020; Borgeaud et al., 2022). This paradigm addresses hallucinations and outdated knowledge in LLMs while enabling source attribution(Lewis et al., 2020). Methods like (Guu et al., 2020; Borgeaud et al., 2022) pretrain retrievers jointly with LLMs, encoding retrieved documents into latent representations for generation. Alternatively, kNN-LM(Khandelwal et al., 2019) interpolate model predictions with retrieved token distributions. While kNN-LM avoids additional training, it still requires access to internal model representations. Recently, the utilization of feedback from LLMs received attention from researchers, (Shi et al., 2023) directly applies LLM probabilities as LLM feedback. While (Wang et al., 2023) introduced a novel approach to iteratively train dense retrievers for identifying high-quality in-context examples, studies have shown that training retrievers to leverage fine-grained LLM feedback significantly enhances in-context learning performance compared to traditional methods like BM25(Reimers, 2019) that do not utilize such feedback.

3 Proposed Method

In this section, we introduce the training pipeline of our method as illustrated in Figure 2, including architecture, training data generation, correlation-enhanced loss, task-mask mechanism.

3.1 Architecture

Retriever We adopt a bi-encoder based dense retriever architecture initialized with $E5_{base}$ due to its excellent performance. Given a query x and the candidate examples $\{c_i\}_{i=1}^n$, our retriever encodes the query x into an embedding $E(x)$ and each of the candidate examples into embeddings $E(c_i)$. The retriever score between the query and each example is computed via the dot product:

$$s(x, c_i) = E(x) \cdot E(c_i) \tag{1}$$

Large Language Model To make a fair comparison with other existing approaches, we opt specifically for LLAMA (Touvron et al., 2023).

3.2 Training data generation

For each training example (x, y), we retrieve top-n candidates $\{(x_i, y_i)\}_{i=1}^n$ from a diverse pool P, excluding (x, y). Candidates are represented as (x_i, y_i), and retrieval is based on x. The candidates are ranked using a frozen LLM by computing the log-likelihood of y given x and each candidate (x_i, y_i):

$$\begin{aligned} P_{LLM}(y|c_i, x) &= Task(\ p_{llm}(y|x, c_i)\ ), \\ \log p_{llm}(y|x, c_i) &= \sum_{j=1}^{n} \log p_{llm}(y_j|x, c_i, y_{<j}), \end{aligned} \tag{2}$$

where $Task()$ assigns a low score if c_i is from a different task than x. This method requires only a single forward pass, making it computationally efficient and task-agnostic.

3.3 Correlation-enhanced Loss

To provide fine-grained supervision for the retriever based on LLM probabilities, we propose a novel **correlation-enhanced loss**. This loss function is designed to align the retriever's behavior with the language model's preferences by explicitly modeling the relationship between retrieval likelihoods and LLM probabilities. In the following, we detail the computation of our proposed loss function.

3.3.1 Probabilities of the retrieved examples

Each candidate example c_i is selected according to its similarity score $s(x, c_i)$ with respect to the query x, where $\{s(x, c_i)\}_{i=1}^n$ represents the set of similarity scores for the top-n candidates. These scores serve as the foundation for computing the retrieval likelihood. Specifically, the retrieval likelihood for each candidate c_i is calculated as:

$$P_R(c_i \mid x) = \frac{e^{s(x,c_i)/\gamma}}{\sum_{c_j \in \mathcal{D}'} e^{s(x,c_j)/\gamma}}, \tag{3}$$

where γ is a hyperparameter that controls the temperature of the softmax. This retrieval likelihood reflects the retriever's confidence in the relevance of each candidate example to the query. Ideally, the retrieval likelihood should be computed by marginalizing over all examples in the corpus $\mathcal{D}$, but this is computationally intractable in practice. Therefore, we approximate the retrieval likelihood by marginalizing only over the retrieved candidate examples $\mathcal{D}'$. And also in our framework, since the retrieval results are pre-computed, we avoid the need to encode the entire corpus during training.

3.3.2 Align probabilities

To align the retriever's behavior with the language model's preferences, we utilize pre-computed LLM probabilities derived from the previously constructed dataset. For each candidate example $c_i \in \mathcal{D}'$, where $\mathcal{D}'$ denotes the set of retrieved candidates, we employ the pre-computed probability $P_{LLM}(y \mid c_i, x)$ as defined in Equation 2. This probability quantifies the likelihood of the ground truth output y given the input context $x \in \mathcal{B}$ and the candidate example c_i. These probabilities are computed using a frozen language model during the dataset construction phase, ensuring consistency and efficiency in training.

The correlation-enhanced loss is defined as the element-wise product of two components: (1) the retrieval likelihood $P_R(c \mid x) \in \mathbb{R}^{n\times m}$, where $n = |\mathcal{B}|$ denotes the batch size and $m = |\mathcal{D}'|$ represents the number of retrieved candidates, and (2) the pre-computed LLM probability $P_{LLM}(y \mid c, x) \in \mathbb{R}^{n\times m}$. Formally, the loss is expressed as:

$$Q_{CE}(c \mid x, y) = P_R(c \mid x) \cdot P_{LLM}(y \mid c, x), \tag{4}$$

This formulation ensures that examples with high LLM probabilities are prioritized during training. The training objective is to optimize the retriever to prioritize candidates with the highest $P_{LLM}(y \mid c, x)$ for better LLM predictions, which is achieved by minimizing the following loss function:

$$\mathcal{L}_{CE} = -\frac{1}{|\mathcal{B}|} \sum_{x\in\mathcal{B}} \sum_{d\in\mathcal{D}'} Q_{CE}(d \mid x, y), \tag{5}$$

where $\mathcal{B}$ is a batch of input contexts. By minimizing this loss, we encourage the retriever to

# of datasets→ task number	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	48.7	80.4	70.4	**72.5**	71.5	59.0	93.6	**49.9**	51.1	66.5
Ours(1 iter)	**55.2**	80.1	64.7	71.3	80.8	**65.0**	92.2	**49.9**	**51.3**	68.0
Ours(2 iter)	55.1	**80.5**	69.1	71.0	81.9	64.3	92.1	49.3	**51.3**	**68.3**
Ours(3 iter)	54.5	79.9	**70.5**	71.5	**82.2**	63.5	90.4	49.0	51.1	68.1

Table 1: Main results on a suite of 30 NLP tasks. Other results come from (Wang et al., 2023).

prioritize examples that are not only relevant to the input context but also beneficial for the language model's predictions.

3.4 Task-Mask Mechanism

$\mathcal{L}_{CE}$ solves the problem of aligning probabilities between our retriever and the LLM, but a crucial issue is observed. Specifically, when calculating $\mathcal{L}_{CE}$, examples from different tasks are inherently assigned very large negative values which results in disproportionately high loss values compared to those from the same task. It aids our retriever in learning to penalize the selection of examples from different tasks, but hinders its ability to find more suitable examples within the same task.

To mitigate this issue, we design a Task-Mask Mechanism that separates the loss computation by introducing loss mask $\mathcal{M} \in \mathbb{R}^{\mathcal{B}}$:

$$\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_{\mathcal{B}}\}$$
$$\mathcal{M}_x = \begin{cases} 1, & \text{if } p_{min} < t \\ 0, & \text{otherwise} \end{cases}, \quad x \in \mathcal{B} \tag{6}$$

Here, t denotes the task threshold, a large negative value, and $\{\}$ signifies the concatenation operation. The term $p_{min} \in \mathbb{R}^1$ denotes the minimum of P_{LLM} with a single batch. $\mathcal{L}_{CE}$ is then divided into two components: the different-task loss $\mathcal{L}_d$, which discourages retrieving from different tasks, and the same-task loss $\mathcal{L}_s$, which encourages retrieving better examples within the same task:

$$\mathcal{L}_d = -\frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \left(\sum_{d \in \mathcal{D}'} Q_{CE}(d \mid x, y) \cdot \mathcal{M}_x \right),$$
$$\mathcal{L}_s = -\frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \left(\sum_{d \in \mathcal{D}'} Q_{CE}(d \mid x, y) \cdot (1 - \mathcal{M}_x) \right), \tag{7}$$

And then in alignment with (Wang et al., 2024), we integrate an InfoNCE-based contrastive loss $\mathcal{L}_{cont}$ (Chen et al., 2020) to incorporate the in-batch negatives by designing the candidate with the highest LLM probabilities as the positive example. Thus, the final training objective for the retriever can be formally expressed as:

$$\mathcal{L}_{retriever} = \lambda \cdot \mathcal{L}_{cont} + \alpha \cdot \mathcal{L}_d + \beta \cdot \mathcal{L}_s \tag{8}$$

where $\{\lambda, \alpha, \beta\}$ are the hyperparameters that determine the relative weighting of the three loss functions.

4 Experiments

4.1 Evaluation Setup

Following the task setting of (Wang et al., 2024), we verify the merit of the proposed TDR for a diverse collection of 30 publicly available NLP tasks(Wei et al., 2021; Cheng et al., 2023; Wang et al., 2024), which span 9 distinct categories and include up to 10k examples per dataset. The training retrieval pool is constructed by combining all training examples, excluding the four datasets QNLI, PIQA, WSC273, and Yelp, aiming to assess the models' generalization ability on unseen tasks. Detailed task classification is shown in Table 2.

Category	Datasets				
Close QA	ARC Challenge	ARC Easy	NQ		
Commonsense	COPA	HellaSwag	**PIQA**		
Coreference	Winogrande	WSC	**WSC273**		
Paraphrase	MRPC	PAWS	QQP		
Sentiment	Sentiment140	SST2	**Yelp**		
Data-to-text	CommonGen	DART	E2E NLG		
Summarize	AESLC	AGNews	Gigaword		
Reading Comp.	BoolQ	MultiRC	OpenBook QA	SQuAD v1	
NLI	MNLI (m)	MNLI (mm)	**QNLI**	RTE	SNLI

Table 2: Detailed datasets used in this paper. The bold texts display four held-out datasets which are unseen during training periods.

#	$\mathcal{L}_{CE}$	Task-Mask	CQA	Comm.	Coref.	NLI	Para.	RC	Sent.	D2T	Summ.	Avg
1			48.0	79.4	**67.0**	67.0	74.0	60.5	91.5	49.6	50.3	65.2
2	✓		54.7	79.6	66.0	71.2	76.4	63.4	91.4	**50.2**	**51.3**	67.3
3	✓	✓	**55.2**	**80.1**	64.7	**71.3**	**80.8**	**65.0**	**92.2**	49.9	**51.3**	**68.0**

Table 3: Ablation study of our proposed TDR on the test set. The values in the table show the average performance of the model across 9 categories consisting of 30 tasks.

During training, we initialize the retriever using the pre-trained $E5_{\text{base}}$ model (Wang et al., 2022). The retriever is fine-tuned on the generated dataset with a batch size of 32 and 4 examples per batch. Training is conducted for 12,000 steps on 8 V100 GPUs, completing in approximately two hours, with a learning rate of 3×10^{-5}. To mitigate the influence of random seeds, we report the average performance metrics across each task category. For task evaluation, we employ LLaMA-7B (Touvron et al., 2023) as the standard language model to ensure consistency and fairness in comparisons. Following prior work (Wang et al., 2023), we retrieve 8 in-context examples for each test input in all evaluations except zero-shot settings.

Building upon this foundation, our method TDR addresses the insufficient utilization of LLM feedback in complex training procedures by explicitly modeling LLM-generated feedback to supervise retriever training. Additionally, we decouple the training of examples across distinct tasks, further enhancing performance across all evaluated tasks. We perform three iterative training cycles, as the second iteration yields the best performance. The experimental results are recorded as "Ours 1 iter," "Ours 2 iter," and "Ours 3 iter" in Table 1. The results demonstrate that our approach achieves significant improvements across seven task categories, delivering an average accuracy gain of 1.8% over the previous state-of-the-art method. Notably, TDR surpasses previous SOTA method by 10.7% on the task category Paraphrase, validating its significant effectiveness.

Furthermore, as shown in Figure 3, our method significantly outperforms the "Random" baseline, achieving an average improvement of 22.2% across all 30 tasks, highlighting its effectiveness in leveraging task - specific information. It also demonstrates robust generalization, consistently beating the random baseline on four unseen training tasks, indicating its ability to handle open - set scenarios. However, it performs relatively poorly on the WSC and RTE tasks, likely due to the limited number of training examples (554 for WSC and 2,490 for RTE) in a 600,000 - example retrieval pool, which may impede the retriever. Despite this, our method still yields competitive results, showing its robustness across diverse tasks.

Detailed experimental results for all 30 tasks are provided in Table 5 of the supplementary material. In the subsequent experiments, we consistently refer to our method as TDR, which corresponds to the "Ours 2 iter" configuration.

5 Analysis

5.1 Ablation Study

Here, we study how each component in TDR influences the overall performance. We consider one or more components at each stage and Table 3 summarizes the results on training set of the 9 categories consisting of 30 NLP tasks. Note that baseline at Row #1 is a dense bi-encoder retriever finetuned by minimize the KL-Divergence between the retriever score distribution and the LLM preference.

By incorporating the $\mathcal{L}_{CE}$ that appropriately aligns the retriever probabilities P_R and task-specific LLM probabilities P_{LLM}, the variant at Row #2 makes the absolute improvement over the base model at Row #1 on the average score. This is not surprised as the correlation-enhanced loss $\mathcal{L}_{CE}$ can establish a positive correlation between the retriever probabilities and the LLM probabilities while avoiding the direct use of KL divergence to fit the distributions, given the significant differences between them. Specifically, the retriever actively adjusts its vector space to bring the example c, which maximizes the probability of the answer y, closer to the given query x.

The task-mask mechanism further enhances the retriever by dividing the training objective into two parts: distinguishing between different tasks and finding better examples within the same task, achieving the best results as shown in Row #3.

5.2 Main Results

Table 1 presents the main results of our experiments. 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.). We adopt "Random" as a benchmark for comparison, which randomly selects examples for in-context learning evaluation. Dense retriever baselines include E5(Wang et al., 2022), SBERT(Reimers, 2019), EPR (Rubin et al., 2021) and LLM-R(Wang et al., 2023).

5.3 Universality and Performance Analysis of TDR

Our method TDR is initially trained using feedback from LLaMA-7B. To validate its universality, we evaluate TDR on the aforementioned dataset in conjunction with larger language models GPT-Neo-2.7B(Black et al., 2021) and LLaMA-13B without training. As shown in Table 4 the results reveal that our method TDR achieves average performance improvements of 0.5% and 1.1% over LLM-R, and surpasses the representative sparse retriever method BM25 by 7.9% and 5.3%, respectively. These findings underscore the versatility of our approach, which seamlessly integrates with diverse LLMs to enhance in-context learning capabilities by retrieving high-quality examples.

Notably, our method exhibits pronounced advantages in task types requiring semantically rich contexts, such as Paraphrase (Para.) and Reading Comprehension (RC) — where retrieved examples exhibit patterns closely aligned with the LLM's response patterns. This performance gain is attributed to the higher-quality context retrieval enabled by our framework. Conversely, tasks of categories like Commonsense Reasoning (Comm.) and Data-to-text (D2T), where retrieved examples diverge significantly from the desired answer patterns and performance relies more heavily on the inherent reasoning capabilities of LLMs, the advantages of our method diminish. This phenomenon is corroborated by Table 1 and Figure 3. Table 6 in the Appendix further illustrates this dichotomy by presenting representative retrieval examples from these two task types.

5.4 Visualization of Training Effects

To evaluate our correlation-enhanced loss, we analyze the retriever's performance before and after training using two metrics: (1) the proportion of retrieved examples from incorrect tasks, and (2) their impact on the language model's output probabilities. The results are shown in Figure 4. In our setup, the retriever retrieves top-40 examples for 10,000 queries. The figure (a) shows the proportion of examples from incorrect tasks decreased from 6.67% to 2.23% after training, demonstrating our loss function's ability to focus on same-task examples. This aligns with our first objective.The figure (b) compares the output probabilities before (blue dots) and after (red dots) training. The red dots are more concentrated in the upper-left triangular region and overall higher, indicating that post-training examples lead to higher probabilities for the correct output y. This is expected, as retrieved examples should maximize y's probabilities, aligning with our second objective.

6 Conclusion

In this work, we address two critical challenges in in-context learning (ICL) for large language models (LLMs): (1) difficulty in distinguishing cross-task data distributions and (2) underutilized fine-grained feedback from LLMs. To tackle these issues, we propose TDR, a novel framework that systematically enhances example retrieval for ICL through

	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
Ours	41.4	67.8	60.4	70.2	82.0	53.4	90.9	46.0	48.8	**62.3**
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
Ours	59.2	83.3	70.4	74.3	82.2	64.6	93.2	49.8	51.9	**69.9**

Table 4: Generalization to LLMs that are not used for training.

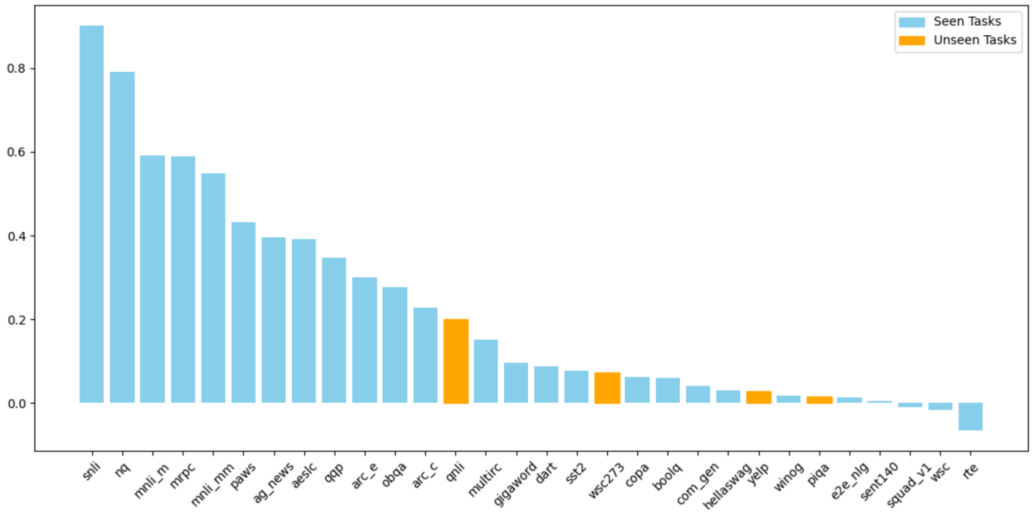

Figure 3: Performance gains of TDR over the random selection baseline.

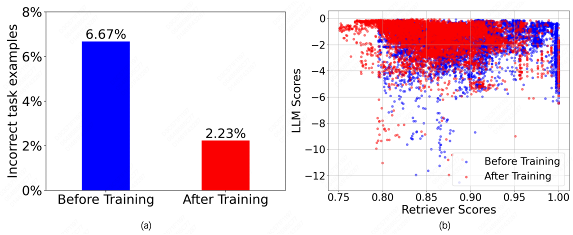

Figure 4: Visualization of Training Effects: (a) Proportion of Cross-Task Retrieval Before and After Training; (b) Correspondence Between Retrieved Examples and LLM Probabilities Before and After Training.

feedback-aware training and task-specific decoupling. The task-decoupled training strategy ensures precise retrieval of domain-relevant examples from multi-task datasets. Simultaneously, by designing a specialized correlation-enhanced loss function to model fine-grained LLM feedback, our method enables retrievers to learn patterns that retrieve better examples for LLMs.

Extensive experiments across 30 diverse NLP tasks demonstrate the superiority of TDR, achieving state-of-the-art performance over existing methods. Notably, our framework shows strong generalization capabilities, maintaining consistent gains on unseen tasks and across LLMs of varying scales. These results validate that explicit modeling of LLM feedback and task-decoupled training strategy are crucial for unlocking the full potential of ICL.

7 Limitations

The inherent feature discrepancies across different tasks presenting persistent challenges in developing task decoupling strategies. In our framework, TDR considers retrieval examples as two categories: belonging to the current task and not belonging to the current task, which may result in the ICL ability not benefiting from examples of similar tasks. More research remains necessary to develop adaptive penalty mechanisms that adjust penalty coefficients based on inter-task feature divergence magnitude, such as applying stronger regularization for tasks with significant feature disparities while reducing constraints for those with minimal discrepancies.

Another limitation of our study is related to the utilization of high-quality examples retrieved during evaluation periods. Based on previous studies, we set the number of in-context examples to 8 and used it for a single round inference evaluation. However, the mutual coordination and influence among retrieval examples, as well as the way in which LLMs utilize these retrieval examples, such as using multiple rounds of evaluation instead, can be a promising direction for further exploration.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58(2).

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

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng 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 arXiv:2303.08518*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

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 arXiv:2212.06713*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

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

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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 arXiv:2305.14128*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. 2024. Iclr: In-context learning of representations. *arXiv preprint arXiv:2501.00070*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

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 arXiv:2212.03533*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

What Makes In-Context Learning Work. Rethinking the role of demonstrations: What makes in-context learning work?

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

A Detailed experimental results

Table 5 shows detailed comparisons between our method and previous in-context example retrieval methods on 30 tasks. On 21 of the tasks, our method achieved state-of-the-art performance and achieved a 1.8% improvement in average performance across all tasks, demonstrating the effectiveness and potential of this in-context example retriever paradigm.

B Pattern analysis of retrieved examples from different task types

As shown in Table 6, for the examples in the two lines above, which come from category Paraphrase (Para.) and Reading Comprehension (RC) respectively, retrieved examples exhibit patterns closely aligned with the patterns of queries and LLM's responses. For the examples in the two lines below, which come from category Commonsense Reasoning (Comm.) and Data-to-text (D2T) respectively, retrieved examples diverge significantly from the desired answer patterns and performance relies more heavily on the inherent reasoning capabilities of LLMs.

C Analysis of retrieval examples from other tasks

As shown in Table 7, examples retrieved from other tasks have similar text content with the queries, but patterns and contents of the retrieved answers are significantly different from those required for the answer corresponding to the query, which makes distinguishing retrieval examples from different tasks an important factor limiting in-context learning performance of LLMs.

Dataset	Zero-shot	Random	Kmeans	BM25	$E5_{base}$	SBERT	EPR	LLM-R	Ours
AESLC	5.8	19.4	19.0	26.8	27.0	25.3	26.0	27.3	27.0
AGNews	31.5	67.4	71.9	90.6	90.6	90.2	91.8	93.5	94.0
ARC Challenge	35.6	39.7	40.5	40.3	44.6	42.8	43.0	43.6	48.8
ARC Easy	51.0	60.0	61.8	59.9	63.0	63.1	63.1	63.3	78.0
BoolQ	64.7	70.0	69.0	74.7	72.4	73.9	74.8	75.1	74.2
CommonGen	19.2	36.3	34.4	37.6	37.4	37.6	39.2	37.7	37.8
COPA	66.0	80.0	85.0	78.0	83.0	82.0	82.0	84.0	85.0
DART	22.9	52.0	46.6	55.9	54.7	54.4	56.2	57.2	56.6
E2E NLG	34.6	52.7	46.4	54.5	51.8	50.2	53.6	54.7	53.4
Gigaword	15.3	30.0	30.7	32.7	32.5	32.6	32.4	32.5	32.9
HellaSwag	71.5	73.9	74.0	74.9	75.2	75.3	75.2	75.5	76.1
MNLI (m)	35.8	46.3	44.2	50.1	44.5	50.8	59.9	70.2	73.7
MNLI (mm)	35.6	48.1	45.4	48.3	44.7	49.3	61.5	72.0	74.5
MRPC	69.1	49.5	38.0	61.8	41.2	52.7	55.9	75.3	78.7
MultiRC	57.0	48.5	34.1	54.2	56.0	55.3	50.4	51.5	55.9
NQ	0.3	21.5	22.6	37.6	39.3	39.4	39.2	39.1	38.5
OpenBook QA	41.6	49.8	49.0	49.6	51.4	51.4	49.6	52.2	63.6
PAWS	53.2	57.0	56.6	56.6	55.4	58.2	57.7	56.6	81.6
PIQA	77.0	79.1	79.4	81.3	81.3	80.7	80.5	81.6	80.3
QNLI	49.2	56.4	53.4	62.2	61.5	61.9	65.0	69.6	67.7
QQP	57.7	63.4	63.3	79.8	77.5	81.3	81.7	82.6	85.4
RTE	59.6	59.9	58.5	65.7	63.9	67.2	66.8	68.6	56.0
Sentiment140	49.3	88.6	89.4	90.8	93.9	92.2	91.4	91.1	89.1
SNLI	39.8	43.7	52.5	47.1	53.5	58.4	68.4	82.0	83.1
SQuAD v1	2.1	64.1	62.3	61.2	60.8	61.6	64.3	57.3	63.5
SST2	54.4	85.9	89.7	84.4	92.1	87.6	88.7	93.8	92.5
Winogrande	62.0	66.7	66.5	67.5	66.9	66.5	66.5	68.1	68.0
WSC	64.4	60.6	56.7	56.7	61.5	63.5	61.5	63.5	79.9
WSC273	74.0	74.4	74.7	64.5	65.2	62.6	65.2	79.5	59.6
Yelp	47.9	92.0	93.5	93.5	97.3	95.9	95.1	95.9	94.7
Average	44.9	57.9	57.0	61.3	61.4	62.1	63.5	66.5	68.3

Table 5: Detailed experimental results for all 30 tasks of our main experiment.

Task name	QQP
Test Input	"How will I contact a good hacker?" "How do l contact a hacker?" Would you say that these questions are the same?
Test Answer	**Yes**
Retrieved Example	"How will I contact a genuine hacker?" "How do l contact a hacker?" Would you say that these questions are the same? **Yes**
Task name	BoolQ
Test Input	Tinker Bell (film series) – A live-action film, with Reese Witherspoon playing Tinker Bell and Victoria Strouse writing the script, is in the works. Can we conclude that are there going to be more tinkerbell movies?
Test Answer	**Yes**
Retrieved Example	Tinker Bell (film series) – A live-action film, with Reese Witherspoon playing Tinker Bell and Victoria Strouse writing the script, is in the works. Can we conclude that are there going to be any more tinkerbell movies? **Yes**
Task name	COPA
Test Input	The horse bucked. What is the cause?
Test Answer	**The rider stroked the horse.**
Retrieved Example	The rider fell to the ground. What is the cause? **The bull bucked the rider.**
Task name	DART
Test Input	Triple: Belgium, LANGUAGE, German language What is a sentence that describes this triple?
Test Answer	**German is the spoken language in Belgium.**
Retrieved Example	Triple: Belgium, LANGUAGE, French language What is a sentence that describes this triple? **French is the spoken language in Belgium.**

Table 6: The bold texts are the ground-truth answers for the test inputs and retrieved candidates. These four examples belong to the category Paraphrase, Reading Comprehension, Commonsense Reasoning and Data-to-text respectively.

Task name	DART
Test Input	Triple: Clowns, priceRange, cheap; Clowns, familyFriendly, yes; Clowns, near, Café Sicilia. What is a sentence that describes this triple?
Test Answer	**A family friendly place is Clowns. It's cheap. It's near Café Sicilia.**
Retrieved Context	Attributes: name = Clowns, priceRange = cheap, familyFriendly = yes, near = Café Sicilia. Produce a detailed sentence about this restaurant.
Retrieved Answer	**A newly-opened venue near Café Sicilia, Clowns offers cheap, family -friendly dining.**
Task name	NQ
Test Input	Question: who do you play as in halo 5? Answer:
Test Answer	**a Spartan**
Retrieved Context	@5toSucceed @halo9 thank you. What is the sentiment of this tweet?
Retrieved Answer	**Positive**
Task name	MultiRC
Test Input	{ { lang } } centers on a man who roams the street night after night. Hidden under his hat and rain jacket he strives for one goal : to find the culprit - the one whom he can make responsible for his suffering . If he wanted to , he could confront him , but he lacks the audacity to do so . He considers suicide , but his courage fails him once again . The options do not appear to present him with a way out and would not personally satisfy him . Finley blames not himself , but only others . In this case he looks to his girlfriend , Violet . He drowns Violet in the bath whilst giving her a massage , Which had become a common ritual for them . On one hand he does this out of malice , on the other to be close to her just one more time. Through this action he wishes to break the growing distance he has come to feel between them , though the actual outcome is the infliction of the greatest possible loneliness , as he turns into a monster . Finley only realizes with hindsight that his misdeeds far surpass those of Violet . Question: "What was Finley doing with Violet before he killed her?" Response: "They were in bed together" Does the response correctly answer the question?
Test Answer	**No**
Retrieved Context	Write a short summary for this text: or how about a girl who is equally obsessed with this guy even though he continually tells her he 's dangerous , could inadvertently kill her and treats her as if she were a child ? this same girl becomes so depressed when her boyfriends breaks up with her that she begins to take risks, some seemingly suicidal , because such behavior summons visions of him.
Retrieved Answer	**some scholars find disturbing elements in twilight books**

Table 7: Retrieved examples from other tasks. The bold texts are the ground-truth answers for the test inputs and retrieved candidates.