

Aligning the Capabilities of Large Language Models with the Context of Information Retrieval via Contrastive Feedback

Qian Dong
dq22@mails.tsinghua.edu.cn
Quan Cheng Laboratory &
DCST, Tsinghua University &
Zhongguancun Laboratory
Beijing, China

Yiding Liu
liuyiding.tanh@gmail.com
Baidu Inc.
Beijing, China

Qingyao Ai*
aiqy@tsinghua.edu.cn
Quan Cheng Laboratory &
DCST, Tsinghua University &
Zhongguancun Laboratory
Beijing, China

Zhijing Wu
zhijingwu@bit.edu.cn
School of Computer Science and
Technology
Beijing Institute of Technology
Beijing, China

Haitao Li
liht22@mails.tsinghua.edu.cn
Quan Cheng Laboratory &
DCST, Tsinghua University &
Zhongguancun Laboratory
Beijing, China

Yiqun Liu
yiqunliu@tsinghua.edu.cn
Quan Cheng Laboratory &
DCST, Tsinghua University &
Zhongguancun Laboratory
Beijing, China

Shuaiqiang Wang
shqiang.wang@gmail.com
Baidu Inc.
Beijing, China

Dawei Yin
yindawei@acm.org
Baidu Inc.
Beijing, China

Shaoping Ma
msp@tsinghua.edu.cn
Quan Cheng Laboratory &
DCST, Tsinghua University &
Zhongguancun Laboratory
Beijing, China

ABSTRACT

Information Retrieval (IR), the process of finding information to satisfy user's information needs, plays an essential role in modern people's lives. Recently, large language models (LLMs) have demonstrated remarkable capabilities across various tasks, some of which are important for IR. Nonetheless, LLMs frequently confront the issue of generating responses that lack specificity. This has limited the overall effectiveness of LLMs for IR in many cases. To address these issues, we present an unsupervised alignment framework called Reinforcement Learning from Contrastive Feedback (RLCF), which empowers LLMs to generate both high-quality and context-specific responses that suit the needs of IR tasks. Specifically, we construct contrastive feedback by comparing each document with its similar documents, and then propose a reward function named Batched-MRR to teach LLMs to generate responses that captures the fine-grained information that distinguish documents from their similar ones. To demonstrate the effectiveness of RLCF, we conducted experiments in two typical applications of LLMs in IR, i.e., data augmentation and summarization. The experimental results

show that RLCF can effectively improve the performance of LLMs in IR context.

KEYWORDS

large language models, information retrieval, reinforcement learning

ACM Reference Format:

Qian Dong, Yiding Liu, Qingyao Ai, Zhijing Wu, Haitao Li, Yiqun Liu, Shuaiqiang Wang, Dawei Yin, and Shaoping Ma. 2018. Aligning the Capabilities of Large Language Models with the Context of Information Retrieval via Contrastive Feedback. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Information Retrieval (IR), which aims to fulfil information needs of peoples though finding relevant documents and knowledge from the Web or large-scale corpus [12], plays a fundamental role in our modern society [10, 20, 58]. Recently, Large Language Models (LLMs) have demonstrated promising performances across a wide range of research fields, including many NLP related tasks such as machine translation and constrained text generation. Yet, despite their advanced capabilities, LLMs are subject to numerous issues such as hallucination and slow knowledge update, which prevent them to serve directly as a reliable information accessing tool. Therefore, IR is still important, attracting a growing number of researchers to investigate the utilization of LLMs to support or empower IR systems. For example, studies [5, 14, 52, 54] have shown that LLMs can create high-quality training data for retrieval models in unsupervised manners by generating queries that are potentially

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXX.XXXXXXX>

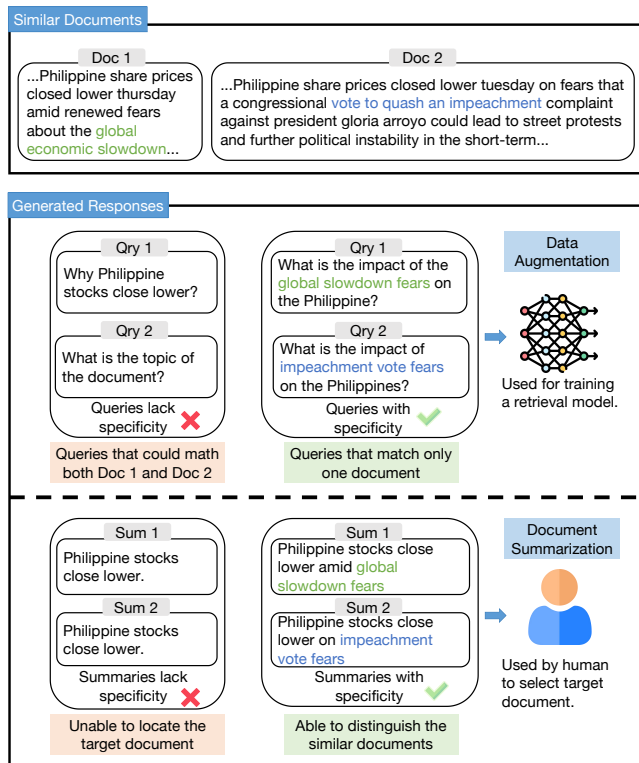


Figure 1: Illustrations of two representative IR application scenarios involving LLMs and comparison between the undesired responses of LLMs and desired ones.

relevant to each document. There are also experiments [29, 36] showing that LLMs can effectively generate meaningful and semantically correct summaries for documents with simple instructions. These studies have demonstrated the significant potential of LLMs in IR.

However, while widely adopted in a wide range of IR research, the paradigm of applying an out-of-the-box LLM directly to IR systems could be suboptimal. Other than the disadvantages mentioned previously, another important problem that limits the effectiveness of LLM in IR context is the misalignment between the capabilities of LLMs and the general needs of IR tasks, particularly the capability to differentiate fine-grained distinctions in documents. In Figure 1, we provide two example cases of two popular applications of LLMs in IR, i.e., data augmentation and document summarization. As depicted in Figure 1, when generating queries or summaries, existing LLMs tend to create reasonable yet unspecific responses that are not differentiable for documents with similar content. However, differentiating similar documents based on fine-grained information is the key to solve numerous IR problems. For example, in search engine, generating unspecific summaries for the retrieved documents (which are inherently similar) make it difficult for users to understand the differences between the documents and locate their desired document [38]. The out-of-the-box LLMs often struggle with this particular capability.

The reason for this lies in that existing training pipeline of LLMs fails to ensure this capability. Typically, the training pipeline of

LLMs undergoes three stages, i.e. *Pre-training*, *Supervised Fine-Tuning (SFT)* and *Alignment*. The **pre-training** stage utilizes next token prediction task to equip LLMs with linguistic knowledge from a massive corpus. The **SFT** stage primarily focuses on stimulating the power of LLMs to support different types of instructions and prompts with supervised data. The **alignment** stage focuses on aligning the capabilities of LLMs with environment feedback. The environment feedback could be from human [41] or a model [3]. The approaches in this stage have shown promise in improving the desired capabilities of LLMs. The pre-training task in the first stage does not require LLMs to distinguish fine-grained distinctions in information, given that subtle token differences merely contribute to a small fraction of the next token prediction loss. Integrating it into LLMs during the SFT stage needs large-scale supervised data, which is challenging and expensive. Consequently, a natural research question is: **Can we align the capability of LLMs with IR context to effectively capture fine-grained distinctions within documents without supervision?**

To this end, we propose a framework called Reinforcement Learning from Contrastive Feedback (RLCF). The whole framework of RLCF is totally unsupervised, thereby eliminating the need for costly labeling expenses. Specifically, we first retrieve a group of similar documents for each document in the corpus through a retrieval model. Then, a group of similar documents is fed to a LLM to obtain a response for each document. The responses could be a query or question related to its document or a summary that captures the main points of the document. Subsequently, we introduce a new metric, Batched-MRR, to evaluate the specificity of response generated by LLM, which is also utilized as reward score in the RLCF framework. Finally, the LLM is optimized by PPO algorithm. The reward score can be considered as contrastive feedback, facilitating the adaptability of LLMs in IR.

To demonstrate the effectiveness of RLCF, we test it in two representative application scenarios of LLMs in IR, i.e., data augmentation and summarization, as shown in Figure 1. The utilization of LLM in data augmentation for IR concentrates on the generating queries to train retrieval models. The application of LLM in the summarization in IR focuses on generating summaries for a list of retrieved documents. In both of these scenarios, specificity in responses is crucial. This specificity helps mitigate the problem of false negatives [14, 44] and enables the precise locating of the desired document respectively. To assess the effectiveness of RLCF-optimized LLMs in data augmentation, we conduct the experiments on BEIR and MS-MARCO. To evaluate the specificity of the generated summaries, we introduce a metric called Rouge-diff and conduct experiments on the LCSTS and Gigaword datasets. We also conduct human evaluation for document summarization. The experimental results demonstrate the effectiveness of RLCF.

We summarize our main contributions as follows:

- We propose a novel framework namely RLCF which utilizes the contrastive feedback to unsupervised align LLMs with IR context.
- We introduce a metric called Batched-MRR for efficient and effective calculation of the reward score in our RLCF framework during training.

- To automatically assess the specificity of responses generated by LLMs, we present a novel metric called Rouge-diff.
- The experimental results demonstrate the effectiveness of our framework. Additionally, we conduct a comprehensive study to examine the effects of RLHF.

2 RELATED WORK

2.1 Large Language Models

Recently, LLMs are emerged and boost considerable natural language processing tasks. The architecture of LLMs, particularly the Transformer [53], leads to significant improvements in capturing both short and long-range dependencies. This advancement has given rise to influential models such as BERT [15] and GPT [45]. These models pave the way for subsequent advancements like GPT-2 [46] and GPT-3 [8], with increasing model sizes and capabilities. The training pipeline of LLMs also earned significant attention in recent years due to its pivotal role in enabling models like GPT to exhibit remarkable language understanding and generation capabilities. Pre-training is a cornerstone of training LLMs and involves training the model on a massive corpus to learn linguistic patterns and structures, leveraging the tasks such as masked language modeling [15], next token prediction [45] and etc. By utilizing large-scale pre-training, LLMs acquire a general understanding of language, making them available for various downstream tasks. Supervised Fine-Tuning (SFT) involves training the LLM on task-specific datasets with labeled examples. This stage adapts the generic linguistic knowledge acquired during pre-training to specific tasks, such as sentiment analysis [18], text classification [17, 19], and dialogues [41]. Alignment technique facilitates LLMs in learning from self-generated responses and environmental feedback, thereby aligning the capability with the desired attribute. The environment feedback could be from human [41] or a model [3]. This approach has shown promise in improving the safety and helpfulness of LLMs.

2.2 Data Augmentation for Dense Retrieval

Dense retrieval plays a crucial role in the field of IR, drawing considerable attention from both academia and industry due to its superior performance in in-domain scenarios [16, 34, 35, 39, 57]. However, numerous studies find that dense retrieval can hardly generalize from one domain to others [14, 52]. Therefore, considerable studies are proposed to tackle this problem. BEIR [52] is a heterogeneous benchmark to evaluate the generalizing ability of retrieval models. Several data augmentation studies are proposed to boost the generalizing ability of retrieval models. Contriever [25] leverages the pseudo query document pairs constructed by heuristic rules to train unsupervised dense retrieval and demonstrates strong performance in BEIR. GenQ [25] is an unsupervised domain-adaption approach for dense retrieval models by training on synthetically data generated by language model. GPL [54] combines a query generator with pseudo labeling from a cross-encoder, yielding significantly improvement over GenQ on the out-of-domain setting. PROMPTAGATOR [14] utilizes a LLM as a few-shot query generator, and creates task-specific retrievers based on the generated data, resulting in a state-of-the-art performance on BEIR. Leveraging LLMs as query generator for data augmentation for dense retrieval is a

widely adopted approach to enhance the generalizing ability of retrieval models. However, the out-of-the-box LLM often generates identical query for similar documents, which is undesired in dense retrieval training due to the false negative issue [55].

2.3 Document Summarization

Document summarization is a vital research area in natural language processing. Numerous approaches are proposed to tackle the task of summarizing documents into short and informative summaries. Here, we provide an overview of document summarization. Extractive summarization methods [28, 37, 40, 62] select sentences or phrases directly from the input document to form a summary. Abstractive summarization [6, 48, 49, 56] approach imitates human that comprehends a source document and writes a summary based on the salient concepts of the document. Multi-document summarization [7, 9, 61] concentrates on generating concise summaries from a cluster of topic-related documents. Besides, PLMs, such as BART [33], GPT-2 [46], and T5 [47], are also be used for multi-document summarization task [1, 42, 51]. In this work, we focus on the document summarization in IR context, in which the model should generate a specific summary for each document in a similar documents group.

2.4 Alignment Techniques

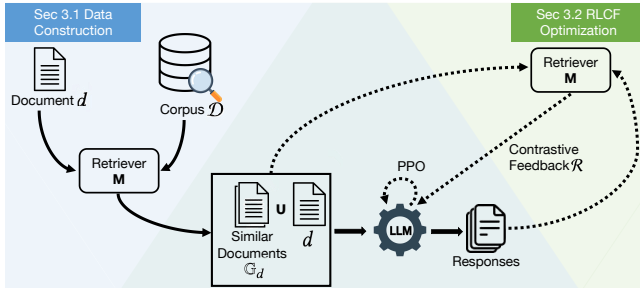
Alignment techniques, which aim to ensure that language models act in accordance with human values or desired intentions, have garnered significant attention. This surge of interest is primarily attributed to the widespread proliferation and increasing impact of language models in recent years. Initial research [11, 21, 40] in this area propose sophisticated heuristic-based reward functions to optimize the language model, enhancing certain properties of the generated responses. In recent years, researchers have increasingly turned their focus to leveraging human feedback as a valuable resource for optimizing language models [4, 50, 64]. Reinforcement learning is typically employed for such optimization, leading to the development of a class of methods referred to as RLHF (Reinforcement Learning from Human Feedback). RLHF leverages human-provided reward signals to guide the training process of language models, enhancing their performance in various natural language generation tasks, such as dialogue [22, 26, 60], translation [2, 30], semantic parsing [32], story generation [63], review generation [13], and evidence extraction [43]. However, a major drawback of RLHF is the requirement for extensive manual labor to provide feedback, making it expensive and time-consuming. To address this limitation, in this work, we propose a novel approach called RLHF (Reinforcement Learning from Contrastive Feedback), which utilizes contrastive feedback to optimize LLMs to generate responses that satisfy the contextual requirements of IR. The RLHF framework eliminates the need for manual labeling, making it more cost-effective.

3 METHOD

In this section, we present the details about our proposed framework, Reinforcement Learning from Contrastive Feedback, dubbed as RLHF. Optimized by RLHF, LLMs could capture fine-grained distinctions in documents, consequently generating specific responses.

Table 1: The notations used in this paper.

Notations	Descriptions
q	The query used to search a document.
d	A document from corpus.
\mathcal{D}	The corpus of documents.
\mathbb{G}_d	The similar document group for document d .
R_d	The response of LLM for document d .
π	The original parameters of LLM.
π_{ϕ}^{RL}	The optimized RL policy.
\mathcal{R}	The reward used to optimize LLM, including the penalty term.
Batched-MRR	A variant MRR used for RLCF optimization.
Inst	An instruction used for few-shot response generation.
Exam	An example for few-shot response generation.

**Figure 2: The framework of RLCF.**

The commonly used notations are summarized in Table 1. The RLCF framework is illustrated in Figure 2. The blue part in this figure represents data construction for contrastive feedback calculation, as elaborated in Section 3.1. The green part in this figure outlines the process of optimizing LLMs through RLCF, with further details provided in Section 3.2.

3.1 Contrastive Data Construction

To avoid the high cost of data labeling, we obtain contrastive feedback through the similar documents groups constructed by a retriever, such as a *dense retrieval model*. These feedback signals contribute to the optimization of LLMs, enabling it to extract fine-grained distinction from similar documents. In this section, we introduce the details about how to construct contrastive data for RLCF optimization.

Similar Documents Retrieval. Since contrastive feedback relies on the comparison of similar documents, the initial step of data construction involves forming groups of similar documents. We utilize a retriever to retrieve similar documents for a given document.

Formally, we retrieve the top- K most similar documents to form the similar document group \mathbb{G} for a given document d in the corpus \mathcal{D} . Each document is limited to the first 512 tokens, and any tokens beyond 512 are truncated. The similar document group \mathbb{G}_{d_i} for document d_i can be defined as

$$\mathbb{G}_{d_i} = \{d_j \mid \underset{\text{top-}K}{\operatorname{argmax}} S(d_i, d_j), \forall d_j \in \mathcal{D}, i \neq j\}. \quad (1)$$

Here, $S(d_i, d_j)$ denotes the similarity score between d_i and d_j , which is defined as

$$S(d_i, d_j) = E_{d_i} \cdot E_{d_j}, \quad (2)$$

where the \cdot means the inner production operation. E_d denotes the embedding of document d , which is the average pooling of the last layer's token representation after the encoding of retriever M , as defined in the following,

$$E_d = \text{Avg_Pooling}(M(d)). \quad (3)$$

Responses Generation. With a similar documents group, we use LLMs to generate responses for each document within the group $\mathbb{G} = \{d \cup \mathbb{G}_d\}$. These responses could be queries or summaries, serving purposes such as data augmentation or summarization.

Formally, we take the concatenation of instruction denoted as *Inst*, several examples denoted as *Exam* and the document $d \in \mathbb{G}$ as input to perform few-shot response generation. The examples *Exam* are randomly chosen from human-labeled data specific to a particular task. This process can be defined as

$$R_d = \text{LLM}(\text{Inst} \oplus \text{Exam} \oplus d). \quad (4)$$

Here, \oplus represents the concatenation operation. Since the search intent of each task for IR is different, we refer to the previous study [14] and design a specific instruction for each task. The prompt templates are presented in Appendix A.

Finally, the document $d \in \mathbb{G}$ and the generated response R_d are utilized in Equation 6 for contrastive feedback calculation, which is then utilized in the subsequent RLCF optimization process.

3.2 RLCF Optimization

Next, we introduce the calculation of contrastive feedback, which serves as the reward score in RLCF. As depicted in Figure 2, the contrastive feedback is calculated by a retriever (e.g., a dense retrieval model) and is a variant of Mean Reciprocal Rank (MRR). MRR is a widely used metric in IR, which is defined as

$$\text{MRR@}K = \frac{1}{|Q|} \sum_{q \in Q} \frac{\mathbf{I}(\text{rank} \leq K)}{\text{rank}}. \quad (5)$$

Here, $\mathbf{I}(\cdot)$ is a indicator function. The *rank* in Equation 5 denotes the position of the *first* relevant document in the retrieved candidates of query q and Q is the query collection of a evaluation set.

Given that the candidates of query q is retrieved from the whole corpus, the traditional MRR is a corpus level metric. Therefore, the computational overhead could be magnified by the scale of massive corpus and eventually cause unacceptable efficiency degeneration of model optimization.

To address this issue, we introduce Batched-MRR, a variant MRR computed in the batch level. Considering that the most similar documents are already grouped in data construction process (as elaborated in Section 3.1) and are subsequently fed into the same batch, Batched-MRR can be viewed as an approximation of MRR, since the documents that most affect the output of indicator function $\mathbf{I}(\cdot)$ in Equation 5 are almost within a batch. The calculation of Batched-MRR can be defined as

$$\text{Batched-MRR}_{R_d} = \frac{1}{\text{rank}}. \quad (6)$$

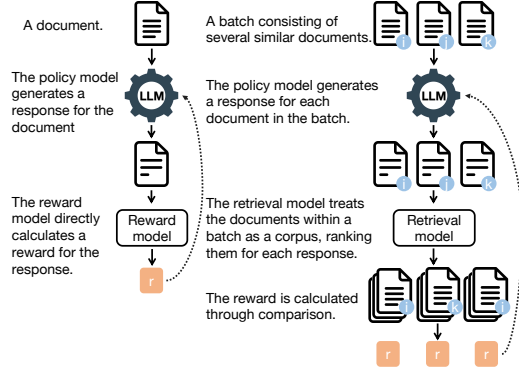


Figure 3: The comparison between RLHF and RLCF.

Here, *rank* represents the position of a document d within a batch, which is sorted by a retriever based on the similarity score $S(d, R_d)$ against the response R_d generated by LLM.

Our objective is to optimize the policy model, i.e., the LLM, using contrastive feedback to generate responses that are desired in the context of IR. We achieve this through reinforcement learning, specifically with the PPO algorithm. We consider the Batched-MRR as the reward score for the entire response, and maximize it using the PPO algorithm. Consistent with prior study [41], we also incorporate a term in the reward that penalizes the KL divergence between the optimized RL policy π_{ϕ}^{RL} with parameters ϕ and the original large language model π . The penalty term prevents the policy model from producing responses that diverge significantly from the vanilla LLM, thereby preserving the language capabilities of the policy model. The full reward \mathcal{R} could be written as

$$\mathcal{R}(d, R_d) = \text{Batched-MRR}_{R_d} - \beta \log \left[\frac{\pi_{\phi}^{RL}(R_d | d)}{\pi(R_d | d)} \right], \quad (7)$$

where β is a hyper parameter that balances the Batched-MRR and penalty term.

Remarks. RLHF [41] demonstrates that it is effective in aligning the outputs of LLMs with human preferences, providing valuable insights into alignment techniques of LLMs. However, RLHF requires a substantial amount of supervised preference data for training a reward model, which is challenging and expensive to get. In the RLCF framework, instead of directly using a reward model [41], we employ a retrieval model to obtain a reward score by comparing documents within a batch. Notably, the documents within the same batch are highly similar. Figure 3 illustrates that rewards in RLCF are obtained at the batch level, whereas in other frameworks like RLHF, rewards are obtained at the single document level. Additionally, the reward model in RLHF demands a substantial amount of human-labeled samples for training to obtain an effective reward score for aligning LLMs with a desired attribute. In contrast, our RLCF framework does not require any additional labeling.

4 EXPERIMENTAL SETUP

4.1 Tasks and Datasets

We utilize two representative application scenarios of LLMs in IR to evaluate the effectiveness of RLCF framework. The first one is data augmentation, and the second one is document summarization, as

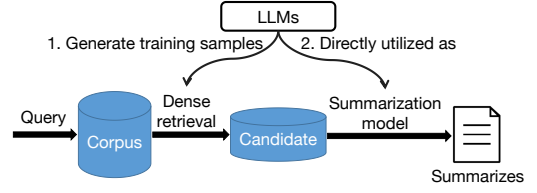


Figure 4: Two representative application scenarios of LLMs in IR.

depicted in Figure 4. The use of LLM in data augmentation for IR primarily focuses on generating queries to train retrieval models using a limited number of examples. The application of LLM in the summarization process within IR is aimed at producing summaries for a list of retrieved documents.

Data Augmentation. There are many diverse and unique retrieval problems, each targeting different search intents, queries, and search domains. Generalizing a retrieval model from one task to another is challenging [52]. It is not feasible to label sufficient training data for each task. Therefore, LLMs could be utilized to augment the training data to alleviate this issue. Queries generated by vanilla LLMs usually suffer from the false negative problem during model training, limiting the effectiveness of data augmentation. Therefore, we propose RLCF to stimulate the LLM to generate specific queries, thereby alleviating this issue. We use the following datasets to evaluate the effectiveness of data augmentation:

- The first one is BEIR [52], which is a widely-used benchmark. BEIR involves 19 datasets, with 15 of them publicly accessible, encompassing 10 IR tasks and 10 domains. Notably, we exclusively select tasks from BEIR with document counts exceeding 10,000 for our experiments, as the documents in a small corpus are trivial to differentiate, and thus is not the focus of this paper. Additionally, the selected tasks need to encompass more than 50 test queries, ensuring a statistical significance.
- Besides, MS-MARCO is also used in a few-shot setting to evaluate the effectiveness of LLMs' data augmentation. MS-MARCO is a large-scale dataset designed for training and evaluating machine learning models in the field of IR and reading comprehension. It was developed by Microsoft Research to address the challenges of real-world information seeking and question-answering tasks.
- We also include the well-known QA datasets, NQ [31] and TriviaQA [27].

Dense retrieval is optimized by the following contrastive loss

$$\mathcal{L}_c = -\log \frac{\exp(S(q, d_+))}{\exp(S(q, d_+)) + \sum_{d_- \in \mathcal{N}_-} \exp(S(q, d_-))}, \quad (8)$$

where \mathcal{N}_- is a set of negative documents (denoted as d_-) for query q . The query q and document d_+ form a positive query-document pair. Following previous studies [14, 25, 54], in data augmentation, the query q generated by LLM for document d form a positive pair. The negative set \mathcal{N}_- consists of the same batch documents [59].

Document Summarization. Another representative application of LLM in IR is document summarization. Summaries generated

Table 2: Experimental results of retrieval methods on BEIR.

Dataset(→)	Question answering			Entity retrieval	Fact checking			Citation prediction	Avg.
	HotpotQA	NQ	Fiqa	DBPedia	Fever	Climate-fever	Scifact	Scidocs	
Model(↓)	NDCG@10								
DPR	39.1	47.4	11.2	26.3	56.2	21.3	31.8	7.7	30.1
BM25	60.3	32.9	23.6	31.3	75.3	14.8	66.5	15.8	40.1
Contriever	48.1	25.4	24.5	29.2	68.2	15.5	64.9	14.9	36.3
DADR	59.0	36.3	33.8	32.0	73.2	17.4	71.1	17.0	42.5
+RLCF	61.0	36.6	34.2	32.3	74.1	17.5	71.7	19.0	43.3
	Recall@100								
DPR	59.1	88.0	34.2	34.9	84.0	39.0	72.7	21.9	51.7
BM25	74.0	76.0	53.9	39.8	93.1	43.6	90.8	35.6	63.4
Contriever	70.4	77.1	56.2	45.3	93.6	44.1	92.6	36.0	64.4
DADR	69.6	80.9	69.2	40.7	92.9	44.4	93.7	40.3	66.5
+RLCF	70.4	80.9	70.1	40.9	92.8	44.0	93.7	41.5	66.8

Table 3: Experimental results of retrieval methods on MS-MARCO, NQ and TriviaQ.

	MS-MARCO	NQ		TriviaQA	
	MRR@10	R@20	R@100	R@20	R@100
BM25	18.7	62.9	78.3	76.4	83.2
Contriever	16.0	62.9	78.3	74.2	83.2
DADR-3B	22.9	72.0	84.3	80.1	85.9
+RLCF	23.3	73.1	84.8	80.3	86.1
DADR-11B	23.1	72.6	84.5	80.4	86.0
+RLCF	23.4	72.6	84.9	80.9	86.6

by vanilla LLMs for similar documents frequently exhibit homogenization, which hampers the user’s capacity to efficiently locate the target document within these summaries. To assess the effectiveness of document summarization for vanilla LLMs and RLCF-optimized LLMs, we perform experiments on two datasets: LCSTS for Chinese and Gigaword for English.

- LCSTS [23] is a widely used dataset for text summarization tasks in Chinese. It was created to facilitate research and development in the field of short text summarization.
- Gigaword [49] is a widely used dataset in the field of text summarization. It is a large-scale collection of news articles and their corresponding headline summaries. This dataset is known for its extensive coverage of diverse topics and its massive size, which makes it a valuable resource for training and evaluating text summarization models.

The corpus of other datasets, such as CNN/Daily Mail, PubMed, and WikiSum, are typically of smaller size, making them relatively easy to distinguish. Therefore, these datasets are not the primary focus of this paper.

4.2 Implementation Details

In our experiments, we employ Flan-T5 as the backbone of LLMs for English datasets, which is an encoder-decoder architecture. We perform experiments using Flan-T5 models with 770M, 3B, and 11B

parameters, respectively. For the Chinese dataset, we utilize BELLE-7B-2M as the backbone of LLMs, which is a decoder-only architecture and achieve promising instruction-following ability in Chinese. Although GPT3.5 and GPT4 demonstrate superior performance, the undisclosed parameters hinder the training and evaluation of RLCF. We use Contriever [25] as the retriever M in RLCF.

Due to the limited computational resources, we employ OpenDelta [24] for parameter-efficient tuning (i.e., LoRA) implementation for all LLMs, except for Flan-T5 with 770M parameters. To maximize the efficient utilization of GPU memory, we optimize all the parameters in Flan-T5 with 770M parameters, the last 23 layers of Flan-T5 with 3B parameters, the last 4 layers of Flan-T5 with 11B parameters, and the last 12 layers of BELLE-7B-2M, respectively. In text generation, we simply use the greedy decoding strategy.

All experiments are implemented with PyTorch and Huggingface. DeepSpeed with ZeRO stage 2 is utilized for efficient training. All the training and evaluation are conducted on 8 NVIDIA Tesla A100 GPUs (with 40G RAM).

4.3 Evaluation

Automatic Evaluation. For data augmentation of dense retrieval, we directly utilize the traditional metrics of document retrieval, namely, Mean Reciprocal Rank (MRR), Recall, and Normalized Discounted Cumulative Gain (NDCG). We introduce Rouge-diff as an evaluation metric for document summarization, aimed at assessing the specificity of summaries within similar documents. The Rouge-diff is a variant of Rouge-N, which is defined as

$$\text{Rouge-diff}_{R_{d_i}} = \frac{|set(R_{d_i}) \cap (set(d_i) \setminus set(\cup \mathbb{G}_{d_i}))|}{|set(d_i) \setminus set(\cup \mathbb{G}_{d_i})|}. \quad (9)$$

Here, $set(t)$ represents the tokens of text t after deduplication, and $|set(t)|$ denotes the number of tokens in $set(t)$. Additionally, we incorporate Batched-MRR as a metric for summarization evaluation. Despite that automatic summarization evaluation methods are efficient, their accuracy requires validation. Hence, we further introduce human evaluation.

Human Evaluation. Firstly, we randomly sample 200 documents. Subsequently, we retrieve the 3 most similar documents for each of

these documents, forming 200 groups documents with 4 documents a group. Finally, we generate summaries for the resulting 200*4 documents using both vanilla LLM and RLCF-optimized LLM. We provide annotation guidelines to our experts and instruct them to conduct a three-level annotation. The details of annotation guidelines are presented in Appendix B. Each annotation pair comprised 4 documents and 8 summaries generated by LLM A and LLM B. The annotation experts are tasked with reading the 4 documents and the 8 summaries to determine which abstract, produced by either LLM A or LLM B, are more effective in distinguishing the 4 similar documents. Vanilla LLM is randomly selected as LLM A or LLM B, while RLCF-optimized LLM is the other.

5 EXPERIMENTAL RESULTS

5.1 Data Augmentation for Dense Retrieval

We perform experiments on various IR tasks, such as question answering, entity retrieval, and fact checking, to evaluate the effectiveness of RLCF-optimized LLMs on **Data Augmentation for Dense Retrieval (DADR)**. As for passage retrieval, we conduct experiments on the widely-used datasets of MS-MARCO, NQ, and TriviaQ. For the remaining tasks, we perform experiments on the heterogeneous IR benchmark, BEIR. In our experiments, all datasets' corpus contain more than 10,000 documents to ensure an adequate number of similar documents for effective contrastive feedback. Consistent with previous studies, MS-MARCO is evaluated using the Mean Reciprocal Rank (MRR@10), while NQ and TriviaQ utilize Recall@20 and Recall@100. For the datasets in BEIR, the evaluation metrics include NDCG@10 and Recall@100.

The experimental results of BEIR is presented in Table 2. The cells highlighted in blue indicate that RLCF-optimized LLM exhibits improvements over vanilla LLM. Given the extensive datasets of the BEIR, we exclusively present the experimental outcomes of Flan-T5-XXL, a LLM with 11 billion parameters, in an effort to streamline computation. From this table, we can draw the following findings:

- Across all tasks, RLCF-optimized LLMs consistently demonstrate enhanced performance on NDCG@10 in data augmentation for dense retrieval. RLCF-optimized LLMs experience a slight decrease in performance on Recall@100 for Fever and Climate-fever, possibly attributed to the distinct nature of fact checking compared to other tasks. Despite that Recall@100 for fact checking shows a decline, the overall performance improves from 66.5 to 66.8. This demonstrates the effectiveness of our RLCF framework for aligning the capability of LLMs with the data augmentation in IR.
- DPR exhibits high performance on the NQ dataset, which is the training set of DPR. However, the performance of DPR declines significantly on different tasks or even the same task with diverse datasets. This highlights the difficulty in generalizing from one task to another. Hence, it is crucial to develop a task-specific dense retrieval model.
- Armed with training samples generated by LLMs, dense retrieval models could undergo substantial enhancement, which aligns with the findings of PROMPTAGATOR [14].

Furthermore, Table 3 displays experimental results for the passage retrieval task. We observe similar trends on these widely-used datasets for passage retrieval as on BEIR. The only exception arises

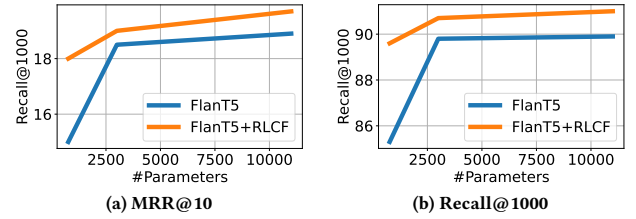


Figure 5: Scaling law of LLMs on data augmentation for dense retrieval.

Table 4: Experimental results of document summarization on LCSTS and Gigaword.

	LCSTS		Gigaword	
	Rouge-diff	Batched-MRR	Rouge-diff	Batched-MRR
LLM	22.1	88.9	11.8	70.9
+RLCF	33.7	90.1	14.3	72.7

in the R@20 metric for the NQ dataset. Despite the lack of significant improvement in R@20, RLCF exhibits substantial improvement in R@100. It is worth noting that if increasing the parameters of the LLMs does not significantly enhance the effectiveness of data augmentation for dense retrieval, then the improvement of RLCF is also marginal.

To further analyze the impact of different parameter counts of LLMs on DADR, we conduct an analysis of the scaling law on MS-MARCO. We use LLMs with parameter counts of 770 million, 3 billion, and 11 billion, respectively. The results are illustrated in Figure 5. The figure demonstrates that the effect of DADR increases with the number of parameters. This is because an increase in the number of parameters leads to higher quality queries generated by the LLMs. As LLMs are trained under a generic domain for query generation, we notice that when the language model has a small number of parameters, a substantial portion of the generated queries resemble the question "What is the main idea of this document?". Additionally, as depicted in Figure 5, the RLCF-optimized LLMs adhere to the scaling law and outperform the LLMs with equivalent parameters.

5.2 Document Summarization

To evaluate the effectiveness of RLCF optimization for LLMs on the document summarization task, we conduct experiments on two datasets, LCSTS and Gigaword, in Chinese and English, respectively. For Chinese document summarization, we employ BELLE-7B-2M, and for English document summarization, we utilize FlanT5-3B as the initial parameters of LLM. We conduct both automatic evaluation and human evaluation for document summarization.

Automatic Evaluation. The automatic evaluation process is elaborately streamlined to improve efficiency. Specifically, we randomly select 512 documents each dataset's corpus to form the initial test set. Subsequently, the four documents most similar to each document in the initial test set are retrieved, thereby extending the

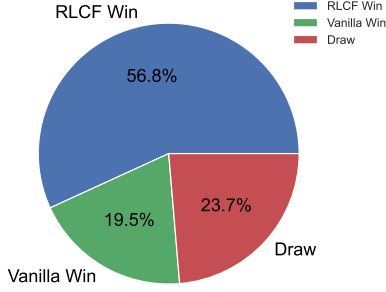


Figure 6: The results of human evaluation.

initial test set and making the evaluation challenging. As a result, the final test set consists of 2048 documents.

The experimental results are presented in Table 4. From this table, we can draw the following findings:

- RLCF optimization significantly improves the Rouge-diff on the test set, demonstrating its effectiveness on document summarization in IR context.
- RLCF optimization leads to significant improvements on both Chinese and English datasets, highlighting its effectiveness across different languages.
- Both Rouge-diff and Batched-MRR metrics on LCSTS are higher compared to Gigaword. This is because LCSTS has fewer documents (2.1 million) compared to Gigaword (3.9 million). Typically, a larger corpus contains more similar documents. As a result, Gigaword presents a more challenging dataset, making it harder to improve Rouge-diff scores.

Human Evaluation. Although automatic summarization evaluation methods are efficient, their accuracy needs validation. Consequently, we also incorporate human evaluation in our experiments. The settings of human evaluation are presented in Section 4.3. The evaluation results are documented in Figure 6. This figure reveals that responses generated by the RLCF-optimized LLM contain more specific information than those produced by the vanilla LLM, making them more suitable for IR contexts.

In order to further examine the distinctions between RLCF-optimized LLMs and vanilla LLMs, we perform case studies in Section 5.3.

5.3 Case Study

In this section, we present several cases to facilitate an intuitive understanding of the effectiveness of RLCF, as shown in Figure 7. For document summarization, we choose three similar documents that focus on the subject of "Philippine stocks close lower". For query generation, we first chose a representative case with a generated query that can be applied to all documents. Subsequently, we chose two similar documents to analyze the their queries generated by vanilla LLMs and RLCF-optimized LLMs, respectively.

For the task of document summarization, we can see that the summaries generated by vanilla LLM are all the same for these similar documents. Despite that the generated summaries are accurate for individual documents, they are not suitable within the pipeline of IR. In the context of IR, once a user submits a query, the search engine retrieves a collection of documents relevant to the query. These

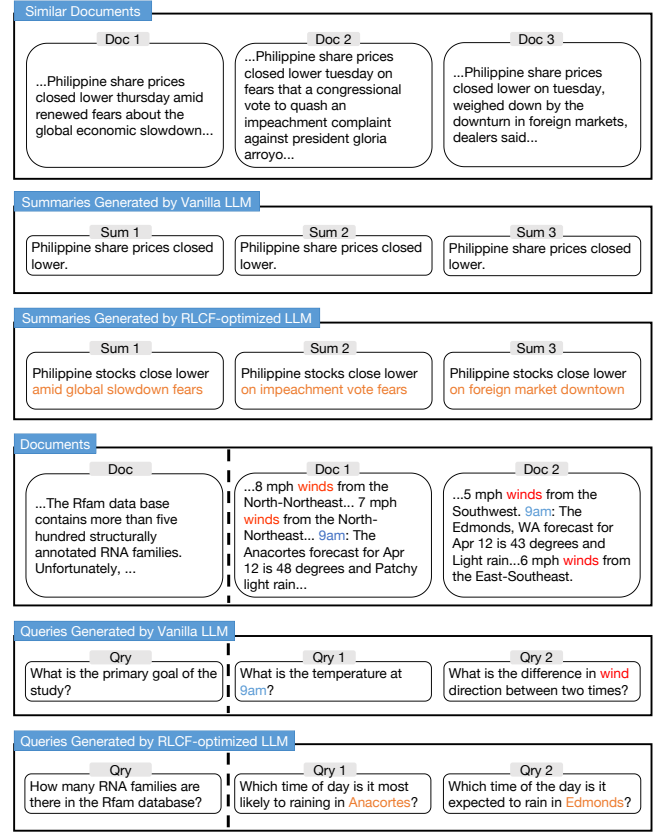


Figure 7: The cases of responses generated by vanilla LLMs and RLCF-optimized LLMs for highly similar documents.

documents naturally possess a high degree of similarity. Generating a distinct summary for such highly similar documents aids users in filtering and identifying their desired documents. As shown in Figure 7, after RLCF optimization, the summaries generated by the LLMs not only precisely summarize the main idea of the document, i.e., the lowering of Philippine stocks, but also provide specific reasons based on their corresponding documents. The summaries generated by RLCF-optimized LLMs demonstrate a higher degree of specificity towards their respective documents, making them more suitable for IR scenarios.

For the task of query generation, it is evident that the queries generated by vanilla LLMs also lack specificity. In the first case of query generation, the query generated by vanilla LLMs could even match all documents. In the second and third cases, despite the generated queries being relatively more relevant to the documents, they still lack specificity. A query generated by vanilla LLMs for one document can still match other documents. Figure 7 demonstrates that the second generated query, "What is the temperature at 9am?", can also be answered by the third document. Likewise, the third generated query solely inquires about changes in wind and can also be answered by the second document. The RLCF-optimized LLMs can accurately generate specific queries based on individual documents. Since the generated query and its corresponding document are positive examples of each other in the contrastive

learning training of dense retrieval, the lack of specificity of the query leads to the false negative problem, which hampers the performance of dense retrieval models [14]. RLCF-optimized LLMs can alleviate this problem and thus improve the effectiveness of data augmentation.

Therefore, through RLCF optimization, the capabilities of LLMs can be effectively aligned with the context of IR, resulting in the generation of more specific summaries and queries for documents

6 CONCLUSION

In this work, we propose a novel framework that leverages contrastive feedback to optimize large language models through reinforcement learning, namely RLCF. The capabilities of LLM could be aligned with the context of information retrieval through the proposed RLCF. Specifically, we first construct a group of similar documents by a dense retrieval model. Subsequently, documents in the same group are fed into a LLM to be optimized. The responses are obtained by the LLM for these similar documents. The contrastive feedback is obtained from these responses generated by LLM with respect to corresponding documents. The contrastive feedback is calculated by dense retrieval model. Formally, we employ a novel reward function, Batched-MRR, as the contrastive feedback, which is a variant of MRR. Then, the contrastive feedback could be utilized to optimized LLM through PPO algorithm, which is a widely used reinforcement learning method. We conduct experiments on two tasks of information retrieval, demonstrating the effectiveness of our proposed RLCF. The RLCF-optimized LLM could generates specific queries for data augmentation, achieving promising performance on few-shot dense retrieval. Besides, we introduce a brand-new setting of document summarization, which is under the context of information retrieval. To be specific, the summarizes should be specific to each document among similar documents, which is desired for users to filter out target document. To evaluate the effectiveness of summarization in the proposed setting, we introduce rouge-diff, a variant of rouge score, which is calculated in the group level. In future work, we suggest exploring more domains which could use the RLCF for optimization, such as style transfer, harmless alignment, helpfulness alignment and etc.

REFERENCES

- [1] Amanuel Alamo, Cori Lohstroh, Erik Madaus, Swati Padhee, Brandy Foster, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Michael Raymer. 2020. Topic-centric unsupervised multi-document summarization of scientific and news articles. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 591–596.
- [2] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086* (2016).
- [3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).
- [4] Florian Böhm, Yang Gao, Christian M Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. *arXiv preprint arXiv:1909.01214* (2019).
- [5] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Data Augmentation for Information Retrieval using Large Language Models. *arXiv preprint arXiv:2202.05144* (2022).
- [6] Mrinmoi Borah, Pankaj Dadure, Partha Pakray, et al. 2022. Comparative analysis of T5 model for abstractive text summarization on different datasets. (2022).
- [7] Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247* (2019).
- [8] 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 (2020), 1877–1901.
- [9] Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.
- [10] Jia Chen, Yiqun Liu, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1524–1534.
- [11] Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080* (2018).
- [12] Anfeng Cheng, Yiding Liu, Weibin Li, Qian Dong, Shuaiqiang Wang, Zhengjie Huang, Shikun Feng, Zhicong Cheng, and Dawei Yin. 2023. Layout-aware Webpage Quality Assessment. *arXiv preprint arXiv:2301.12152* (2023).
- [13] Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiujun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. 2018. Towards coherent and cohesive long-form text generation. *arXiv preprint arXiv:1811.00511* (2018).
- [14] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755* (2022).
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [16] Qian Dong, Yiding Liu, Qingyao Ai, Haitao Li, Shuaiqiang Wang, Yiqun Liu, Dawei Yin, and Shaoping Ma. 2023. I³ Retriever: Incorporating Implicit Interaction in Pre-trained Language Models for Passage Retrieval. *arXiv preprint arXiv:2306.02371* (2023).
- [17] Qian Dong, Yiding Liu, Suqi Cheng, Shuaiqiang Wang, Zhicong Cheng, Shuzi Niu, and Dawei Yin. 2022. Incorporating Explicit Knowledge in Pre-trained Language Models for Passage Re-ranking. *arXiv preprint arXiv:2204.11673* (2022).
- [18] Qian Dong and Shuzi Niu. 2021. Latent Graph Recurrent Network for Document Ranking. In *International Conference on Database Systems for Advanced Applications*. Springer, 88–103.
- [19] Qian Dong and Shuzi Niu. 2021. Legal judgment prediction via relational learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 983–992.
- [20] Qian Dong, Shuzi Niu, Tao Yuan, and Yucheng Li. 2022. Disentangled Graph Recurrent Network for Document Ranking. *Data Science and Engineering* 7, 1 (2022), 30–43.
- [21] Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. *arXiv preprint arXiv:1809.09672* (2018).
- [22] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415* (2019).
- [23] Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865* (2015).
- [24] Shengding Hu, Ning Ding, Weilin Zhao, Xingtai Lv, Zhen Zhang, Zhiyuan Liu, and Maosong Sun. 2023. OpenDelta: A Plug-and-play Library for Parameter-efficient Adaptation of Pre-trained Models. *arXiv preprint arXiv:2307.03084* (2023).
- [25] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).
- [26] Natasha Jaques, Asma Ghandeharion, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456* (2019).
- [27] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551* (2017).
- [28] Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys* 55, 8 (2022), 1–35.
- [29] Bonan Kou, Muhao Chen, and Tianyi Zhang. 2023. Automated Summarization of Stack Overflow Posts. *arXiv preprint arXiv:2305.16680* (2023).
- [30] Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? *arXiv preprint arXiv:1804.05958* (2018).
- [31] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.
- [32] Carolin Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. *arXiv preprint*

- arXiv:1805.01252 (2018).
- [33] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).
- [34] Canjia Li, Xiaoyang Wang, Dongdong Li, Yiding Liu, Yu Lu, Shuaiqiang Wang, Zhicong Cheng, Simiu Gu, and Dawei Yin. 2023. Pretrained Language Model based Web Search Ranking: From Relevance to Satisfaction. *arXiv preprint arXiv:2306.01599* (2023).
- [35] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. *arXiv preprint arXiv:2304.11370* (2023).
- [36] Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848* (2023).
- [37] Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318* (2019).
- [38] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. *arXiv preprint arXiv:2106.03373* (2021).
- [39] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, et al. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *arXiv preprint arXiv:2205.09153* (2022).
- [40] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636* (2018).
- [41] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.
- [42] Richard Yuanzhe Pang, Adam D Lelkes, Vinh Q Tran, and Cong Yu. 2021. AgreeSum: Agreement-oriented multi-document summarization. *arXiv preprint arXiv:2106.02278* (2021).
- [43] Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and Kyunghyun Cho. 2019. Finding generalizable evidence by learning to convince q&a models. *arXiv preprint arXiv:1909.05863* (2019).
- [44] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5835–5847.
- [45] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).
- [46] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
- [47] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.
- [48] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).
- [49] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* (2017).
- [50] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33 (2020), 3008–3021.
- [51] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J Barezi, and Pascale Fung. 2020. CAIRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. *arXiv preprint arXiv:2005.03975* (2020).
- [52] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).
- [53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [54] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577* (2021).
- [55] Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zijiang Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, et al. 2023. T2Ranking: A large-scale Chinese Benchmark for Passage Ranking. *arXiv preprint arXiv:2304.03679* (2023).

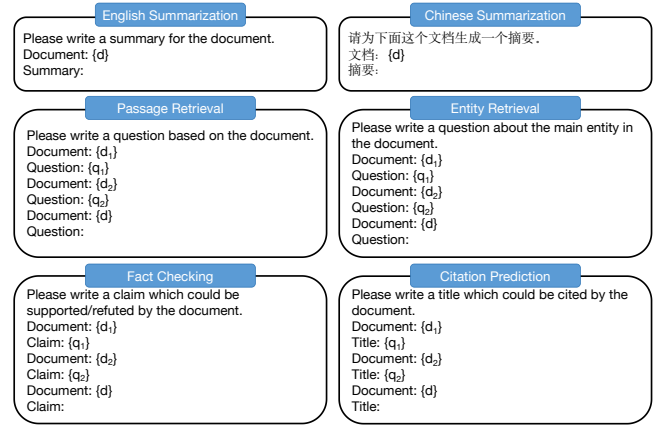


Figure 8: The templates used in our RLCF framework.

- [56] Hemant Yadav, Nehal Patel, and Dishank Jani. 2023. Fine-Tuning BART for Abstractive Reviews Summarization. In *Computational Intelligence: Select Proceedings of InCITE 2022*. Springer, 375–385.
- [57] Shenghao Yang, Haitao Li, Zhumin Chu, Jingtao Zhan, Yiqun Liu, Min Zhang, and Shaoping Ma. [n. d.]. THUIR at the NTCIR-16 WWW-4 Task. ([n. d.]).
- [58] Shenghao Yang, Yiqun Liu, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Enhance Performance of Ad-hoc Search via Prompt Learning. In *China Conference on Information Retrieval*. Springer, 28–39.
- [59] Wenwen Ye, Yiding Liu, Lixin Zou, Hengyi Cai, Suqi Cheng, Shuaiqiang Wang, and Dawei Yin. 2022. Fast Semantic Matching via Flexible Contextualized Interaction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1275–1283.
- [60] Sanghyun Yi, Rahul Goel, Chandra Khatri, Alessandra Cervone, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015* (2019).
- [61] Xin Zheng, Aixin Sun, Jing Li, and Karthik Muthuswamy. 2019. Subtopic-driven multi-document summarization. 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)*. 3153–3162.
- [62] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305* (2018).
- [63] Wangchunshu Zhou and Ke Xu. 2020. Learning to compare for better training and evaluation of open domain natural language generation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9717–9724.
- [64] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593* (2019).

A PROMPT TEMPLATE

The templates used in our RLCF framework are summarized in Figure 8 for reference. Each template begins with an instruction, outlining the tasks requirement of the LLMs. Considering the diversity of IR tasks, we offer up to two examples to assist LLMs in adapting to a specific task. Similar to prior studies, the number of examples depends on the length of the document d due to the input length restriction of LLMs. As shown in Figure 8, d_1 and q_1 , d_2 and q_2 , are two examples of documents and their corresponding queries.

B ANNOTATION GUIDELINE

The annotation guidelines involve three dimensions: specificity, correctness and concision.

- **Specificity.** Can the summary be distinguished from similar documents?

- **Correctness.** Is the summary correct and complete?
- **Concision.** Whether the summary is concise?

The annotation process in RLCF is conducted at the group level, wherein the ultimate decision regarding superior responses is made through comprehensive evaluation.