REAR: A Relevance-Aware Retrieval-Augmented Framework for Open-Domain Question Answering

Yuhao Wang[1*] **Ruiyang Ren**[1*] **Junyi Li**[1,3] **Wayne Xin Zhao**[1†]
Jing Liu[4†] **Ji-Rong Wen**[1,2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]DIRO, Université de Montréal [4]Baidu Inc.
{yh.wang, reyon.ren, lijunyi, jrwen}@ruc.edu.cn, batmanfly@gmail.com

Abstract

Considering the limited internal parametric knowledge, retrieval-augmented generation (RAG) has been widely used to extend the knowledge scope of large language models (LLMs). Despite the extensive efforts on RAG research, in existing methods, LLMs cannot precisely assess the relevance of retrieved documents, thus likely leading to misleading or even incorrect utilization of external knowledge (*i.e.,* retrieved documents). To address this issue, in this paper, we propose **REAR**, a **RE**levance-**A**ware **R**etrieval-augmented approach for open-domain question answering (QA). As the key motivation, we aim to enhance the self-awareness of source relevance for LLMs, so as to adaptively utilize external knowledge in RAG systems. Specially, we develop a new architecture for LLM based RAG system, by incorporating a specially designed rank head that precisely assesses the relevance of retrieved documents. Furthermore, we propose an improved training method based on bi-granularity relevance fusion and noise-resistant training. By combining the improvements in both architecture and training, our proposed REAR can better utilize external knowledge by effectively perceiving the relevance of retrieved documents. Experiments on four open-domain QA tasks show that REAR significantly outperforms previous a number of competitive RAG approaches. Our code and data can be accessed at `https://github.com/RUCAIBox/REAR`.

1 Introduction

Despite the excellent capacities, large language models (LLMs) (Brown et al., 2020; Zhao et al., 2023) still struggle with knowledge-intensive tasks like open-domain question answering (QA), lacking in real-time and domain knowledge (Li et al., 2023a). To mitigate this issue, retrieval-augmented generation (RAG) provides LLMs with potentially relevant documents through retrieval (Gao et al., 2023), aiding in generating more precise content.

While RAG offers clear benefits, it also introduces several technical challenges for effectively improving LLMs. Firstly, the retrieved results likely contain irrelevant content or documents, which may mislead LLMs and even cause them to respond incorrectly (Mallen et al., 2023; Ren et al., 2023). Moreover, it has become common to incorporate multiple reference documents to boost the overall reliability of retrieved documents, whereas this approach potentially amplifies the impact of the noise present in the retrieved documents (Liu et al., 2023; Shi et al., 2023). Thus, LLMs face difficulties in filtering irrelevant documents and integrating their internal knowledge, which needs to avoid potential interference with noisy content.

Recently, several studies (Asai et al., 2023; Luo et al., 2023; Yoran et al., 2023) have attempted to enhance the robustness of RAG systems. For instance, Self-RAG (Asai et al., 2023) allows the model to introspect its outputs by generating special tokens to discriminate if the documents are relevant, and RobustLM (Yoran et al., 2023) prompts LLMs to first discriminate if the documents are relevant and then generate answers. However, these approaches perform the assessment of document relevance solely based on binary labels, which are highly sparse and not precise to capture the fine-grained relevance. In addition, they seldom consider the varied relevance degree of reference documents, making the utilization somehow blind.

To this end, in this paper, we propose **REAR**, a **RE**levance-**A**ware **R**etrieval-augmented approach for open-domain question answering (QA). The key idea of our approach is to develop strong self-awareness of source relevance for RAG systems, so that the LLM can learn to adaptively utilize external knowledge (*i.e.,* retrieved documents) for solving complex QA tasks. To achieve this goal, we make

*Equal contributions.
†Corresponding authors.

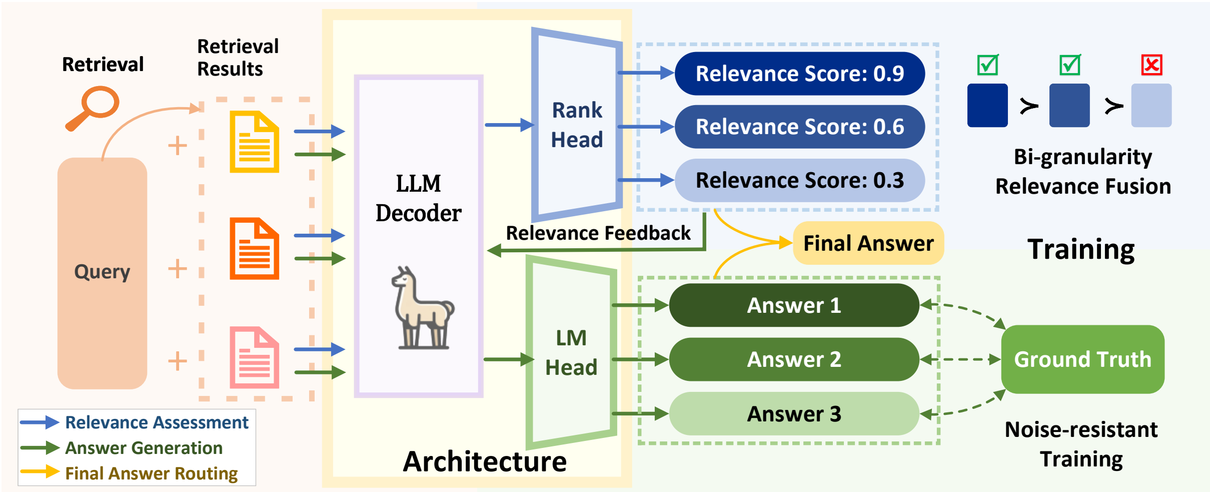

Figure 1: The overview of the proposed REAR framework.

two major contributions in both model architecture and training. Firstly, inspired by the success of reranker in the field of information retrieval, which can discern the differences in relevance among retrieved documents with fine-grained preference (Zhao et al., 2023), we fuse LLMs with a specially designed rank head for relevance assessment, and then employ it to capture relevance signals, to avoid distractions from irrelevant external knowledge. Secondly, to overcome the limitations of binary discriminative methods that can only provide coarse-grained supervision signals, we further design bi-granularity relevance fusion by additionally leveraging fine-grained supervision, and also enhance the discrimination ability of RAG systems by noise-resistant training.

To summarize, our proposed framework enhances the self-awareness of source relevance when using reference documents in RAG systems, with technical improvements on both model architecture and training. Extensive experiments across four different open-domain QA tasks demonstrate the effectiveness of our REAR framework.

2 Related Work

Open-domain Question Answering. Modern open-domain QA systems combine traditional IR techniques with neural reading comprehension models (Chen et al., 2017). After retrieving documents (Ren et al., 2021a; Zhang et al., 2021), an extractive or generative reader is typically used for answer generation (Zhu et al., 2021). Models like REALM (Guu et al., 2020), RAG (Lewis et al., 2020), RETRO (Borgeaud et al., 2022) and In-context RALM (Ram et al., 2023) have demonstrated improved factual generation capabilities. However, these readers make generation quality more prone to noise impact, for lacking explicit relevance discernment. We propose an architecture that explicitly generates relevance scores to assist in subsequent generation tasks.

Retrieval-augmented LLMs. Several research aims at aligning the retriever outputs with the preferences of the LLMs (Izacard and Grave, 2021a; Sachan et al., 2021). And works like Atlas (Izacard et al., 2022), RA-DIT (Lin et al., 2023) jointly train the language model and the retriever for advanced performance on RAG. Some other work improves the quality of retrieved documents by expanding the knowledge sources (Li et al., 2023b) or query rewriting (Zheng et al., 2023). However, we focus on a scenario where the irrelevant documents from retrieval could mislead LLMs. Several recent studies (Asai et al., 2023; Luo et al., 2023; Yoran et al., 2023) attempt to adopt a paradigm in which an initial judgment on relevance is made by generating a statement or special token before proceeding to content generation. However, these methods still lack accuracy in relevance discrimination and LLMs are still vulnerable to irrelevant document interference. Therefore, we propose a framework that can accurately assess the relevance degree, and is more robust to irrelevant content.

3 Task Formulation

In this work, we focus on the task of open-domain question answering (QA) (Chen et al., 2017; Zhao et al., 2024), aiming at answering questions using a large collection of documents. Typically, open-domain QA tasks are often tackled with a *retriever-reader* approach (Chen and Yih, 2020), where the retriever finds relevant evidence and the reader generates the answer based on the retrieved evidence.

Formally, given a query q, the retriever outputs top-k documents $\mathcal{D} = \{d_i\}_{i=1}^k$ from a document collection (can be refined by an optional *reranker*) at the first stage. Different from prior work that combines the entire set of retrieved documents as a unified reference for answer generation (Hofstätter et al., 2023; Luo et al., 2023; Xu et al., 2023), our approach emphasizes individual document utilization, which can be also extended to a multi-document setting. Given the input query q and reference documents $\mathcal{D} = \{d_i\}_{i=1}^k$, the reader (*i.e.,* the LLM) generates an answer a_i based on each reference document d_i, forming an answer set $\mathcal{A}$:

$$\mathcal{A} = \{a_i\}_{i=1}^k = \{\text{LLM}(q, d_i) \mid d_i \in \mathcal{D}\}. \quad (1)$$

Subsequently, we can choose the final answer from $\mathcal{A}$ based on some specific ways, ensuring it aligns best with the query q.

Based on this task formulation, we consider enhancing two key aspects: precise evaluation of relevance between queries and documents (*identifying relevant references*), and leveraging relevance signal for noise-resistant generation (*reducing the influence of irrelevant content*). Therefore, we introduce a relevance-aware approach designed specifically for these challenges.

4 Methodology

In this section, we present the proposed **RE**levance-**A**ware **R**etrieval-augmented generation framework (**REAR**), which is capable of precisely assessing the relevance between queries and documents by incorporating a reranker module within the LLM. Furthermore, we propose improved training methods to enhance the performance of the entire framework, including bi-granularity relevance fusion and noise-resistant training.

4.1 Relevance-Aware RAG Architecture

In this part, we propose a novel architecture that augments the LLM with a ranking module for enhancing the awareness of relevance. As shown in Figure 1, the architecture encompasses three parts, including relevance assessment, relevance-guided generation, and final answer routing.

4.1.1 Relevance Assessment

Instead of treating the retrieved documents equally, we first aim to assess the relevance scores of query-document pairs with the incorporation of a reranker module. Drawing from the success of rerankers in achieving precise relevance assessment (Ma et al., 2023; Sun et al., 2023), we introduce a *rank head* upon the LLM, designed for capturing relevance signals through full interactions between queries and documents. For assessing the relevance score, the LLM decoder maps the input query-document pair into an embedding $\boldsymbol{v}_{\text{rel}}$:

$$\boldsymbol{v}_{\text{rel}} = \text{Decoder}(q, d). \quad (2)$$

Subsequently, $\boldsymbol{v}_{\text{rel}}$ is quantified into a score s_{rel} by a specially designed rank head:

$$s_{\text{rel}} = \text{RankHead}(\boldsymbol{v}_{\text{rel}}), \quad (3)$$

where $\text{RankHead}(\cdot)$ is implemented as a linear projection layer. This design offers two advantages. Firstly, the rank head specializes in relevance assessment and does not affect the functionalities of other modules in the LLM. Secondly, the relevance score s_{rel} can be directly optimized based on the internal states of the LLM with various loss functions, such as contrastive learning and preference optimization, making the training of relevance assessment more flexible and concentrated (will be introduced in Section 4.2).

4.1.2 Relevance-guided Generation

Different from previous work that ignores the relevance of document (Cuconasu et al., 2024), we aim to integrate the relevance assessment score of each document into LLMs to guide the answer generation process. Since the relevance score s_{rel} (in Eq. 3) is a scalar, which may not be fully utilized by LLMs, we further incorporate an embedding layer to map it into a dense vector $\boldsymbol{v}_{\text{guide}}$ as:

$$\boldsymbol{v}_{\text{guide}} = \text{Embedding}(s_{\text{rel}}). \quad (4)$$

This embedding vector serves as a cue for the LLM to generate an answer a based on either the internal knowledge of LLM (the relevance score is low) or external evidence (the relevance score is high) as:

$$a = \text{LLM}(q, d, \boldsymbol{v}_{\text{guide}}). \quad (5)$$

Note that we do not directly utilize the dense vector $\boldsymbol{v}_{\text{rel}}$ (Eq. 2) as the signal $\boldsymbol{v}_{\text{guide}}$ to guide the answer generation (Eq. 5). The reasons are twofold. First,

$\boldsymbol{v}_{\text{rel}}$ plays the role of query-document relevance assessment, while $\boldsymbol{v}_{\text{guide}}$ is supposed to guide LLMs to generate correct answers. Thus, employing different relevance representations $\boldsymbol{v}_{\text{rel}}$ and guide signal $\boldsymbol{v}_{\text{guide}}$ can avoid the interference between relevance assessment and answer generation. Second, our experiments show that utilizing distinct $\boldsymbol{v}_{\text{rel}}$ and $\boldsymbol{v}_{\text{guide}}$ in our framework leads to a superior performance (see Section 5.3.1).

4.1.3 Final Answer Routing

After the retrieval process, we can obtain the top-k relevant documents $\mathcal{D} = \{d_i\}_{i=1}^k$. Based on Eq. 2-Eq. 5, we conduct inference on each query-document pair (q, d_i) to obtain corresponding answer a_i, ended up with a set of k answers $\mathcal{A} = \{a_i\}_{i=1}^k$. We view the generation process from different retrieved documents as *navigating distinct reasoning paths*, and we aim to select the most reliable routing path to obtain the final answer. Specially, we propose two routing strategies, namely path-reliability and knowledge-consistency.

- *Path-reliability routing*: A straightforward approach is to select the answer that the LLM deems to have the highest relevance score:

$$a^* = \arg\max_{a_i \in \mathcal{A}} s_{\text{rel}}(q, d_i), \tag{6}$$

where $s_{\text{rel}}(q, d_i)$ represents the relevance score of pair (q, d_i) obtained by Eq. 3.

- *Knowledge-consistency routing*: Inspired by the success of self-consistency in Chain-of-Thought reasoning (Wang et al., 2022; Wei et al., 2022), this approach aims to select the answer that is based on the *external knowledge* (*i.e.,* retrieved documents), which is highly consistent with the LLMs' *internal knowledge* (*i.e.,* parametric knowledge). In particular, we set the relevance score to zero (denoted by $\hat{s}_{\text{rel}}$) and calculate the inverse of perplexity c_i (Meister and Cotterell, 2021) of generating the answer a_i as follows:

$$c_i = \frac{1}{\text{PPL}(a_i \mid q, d_i, \hat{s}_{\text{rel}} = 0)}. \tag{7}$$

By setting the relevance score to zero, the LLM tends to generate the answer a_i based on its internal knowledge instead of the retrieved document d_i. If the answer a_i can also be generated with lower perplexity (higher knowledge-consistency score c_i) when $\hat{s}_{\text{rel}} = 0$, it indicates that the external knowledge is highly consistent with the internal knowledge. Thus, we first select a subset of documents with high relevance by a threshold γ:

$$\mathcal{D}^+ = \{d_i \mid s_{\text{rel}}(q, d_i) > \gamma\}. \tag{8}$$

Then, we linearly combine the knowledge-consistency score c_i with the relevance score $s_{\text{rel}}(q, d_i)$ with a hyperparameter λ to select the final answer, denoted as a_c^*:

$$a_c^* = \arg\max_{\{a_i \mid d_i \in \mathcal{D}^+\}} \left(s_{\text{rel}}(q, d_i) + \lambda c_i \right). \tag{9}$$

4.2 Model Training

In this part, we will introduce the training method for optimizing our approach.

4.2.1 Bi-granularity Relevance Fusion

Precise relevance assessment is crucial for the reliable utilization of retrieved documents. Previous work often adopts the coarse-grained binary discrimination task (Yoran et al., 2023), which cannot provide sufficient evidence for solving complex QA tasks. Therefore, we consider further incorporating a fine-grained ranking optimization objective, to improve the model training.

Specially, for the coarse-grained supervision, we set the binary labels for candidate documents $\mathcal{D} = \{d_i\}_{i=1}^k$, denoting "*irrelevant*" and "*relevant*" as $y \in \{0, 1\}$, and optimize the following loss:

$$\mathcal{L}_{\text{coarse}} = -\sum_{i=1}^{k} y_i \log(\sigma_i) + (1 - y_i) \log(1 - \sigma_i), \tag{10}$$

where σ_i denotes the normalized probability of assessing (q, d_i) as relevant by the LLM.

For the fine-grained supervision, we utilize the estimated relevance scores s_{rel} for deriving ranking preference constraints:

$$\mathcal{L}_{\text{fine}} = -\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \log(\sigma_i - \sigma_j). \tag{11}$$

Note that we assume that the complete annotation of partial preference relations for the retrieved documents $\mathcal{D} = \{d_i\}_{i=1}^k$ is available during training (See Section 4.2.3), where a smaller index indicates a more relevant document. Furthermore, we combine these two loss functions, as the objective of the bi-granularity relevance fusion:

$$\mathcal{L}_{\text{bi-granularity}} = \mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{fine}}. \tag{12}$$

4.2.2 Noise-resistant Training

In addition to improving the capability of identifying relevant documents, we further consider enhancing the discrimination ability when reference documents contain irrelevant content or even noise, such that the LLM can adaptively use external evidence for task solving. Specially, we further incorporate negative example documents $\mathcal{D}^-$ into the original corpus $\mathcal{D}$ for optimizing LLMs:

$$\mathcal{L}_{\text{noise-resistant}} = \sum_{d_i \in \mathcal{D} \cup \mathcal{D}^-} \log P(a \mid q, d_i, s_{\text{rel}}). \tag{13}$$

Through noise-resistant training, the LLM can learn to discern the incorporation of irrelevant documents, without being encumbered by extraneous information. It enforces LLMs to judge the relevance of external evidence, so as to decide whether to use the retrieved documents for answer generation.

Finally, we define the overall loss function for our REAR framework by combining the bi-granularity loss by Eq. 12 and noise-resistant loss by Eq. 13:

$$\mathcal{L}_{\text{REAR}} = \mathcal{L}_{\text{bi-granularity}} + \mathcal{L}_{\text{noise-resistant}}. \tag{14}$$

4.2.3 Training Data Construction

To optimize our model, we need high-quality training data (both positive and negative samples), and next introduce the concrete strategy for training data construction.

Relevance Labels Acquisition. To obtain fine-grained relevance labels used in Section 4.2.1, we employ a small-scale reranker with cross-encoder architecture to acquire the relevance assessments. In the field of information retrieval, the cross-encoder architecture is regarded as effective for assessing relevance degree, as it considers the query-document pair as a single text, and processes it via an encoder to obtain relevance scores s_{ce} through full interactions (Santhanam et al., 2022; Zhao et al., 2024). In combination with the traditional method of binary annotating label y, the generated score is given as:

$$s_{\text{rel}} = \frac{1}{2}\left(s_{\text{ce}} + y\right), \tag{15}$$

where $y \in \{0, 1\}$ is the binary relevance label and $s_{\text{ce}} \in [0, 1]$ is the continuous scores assessed by the aforementioned cross-encoder.

Irrelevant Documents Sampling. The training method necessitates the use of irrelevant (negative) documents. It has been shown that negative sampling has a large impact on relevance assessment (Xiong et al., 2020). Specially, we refine SimANS (Zhou et al., 2022) that ensures negatives are neither too difficult (false negatives) nor too trivial (uninformative):

$$p_i \propto \begin{cases} \exp(-a(s_i - \hat{s}^+ - b)^2), & s_i < \hat{s}^+ - b, \\ \exp(-ak(s_i - \hat{s}^+ - b)^2), & s_i \geq \hat{s}^+ - b, \end{cases} \tag{16}$$

where the sampling probability for the hard negative document is p_i, s_i and $\hat{s}^+$ respectively denote the relevance scores of document d_i and the positive document, and a, b, and k are hyperparameters. As the relevance scores are evaluated by rerankers, the likelihood of high-scoring samples being false negatives increases. By incorporating a decay scaler k into the sampling probability when relevance scores are high, we reduce the chance of sampling false negatives.

4.3 Model Discussion

Our primary contribution lies in the integration of a special ranking module (*i.e., rank head* in Section 4.1.1) with LLMs. In this way, relevance assessment and answer generation can be approached by the rank head and generation head, respectively, which can reduce the interference between the two different objects. Thus, this method more effectively assesses the fine-grained query-document relevance and further guides the answer generation.

Another key improvement is our training strategy. The bi-granularity relevance fusion enables the LLM to identify not just the overall relevance degree of documents but also to discriminate the relevance preference among the candidates. Consequently, our REAR shows superior performance in evaluating single documents and choosing among multiple documents. In noise-resistant generation training, the LLM is specifically taught to leverage relevance scores for the selective utilization of retrieved documents. This process fosters a dynamic capability to utilize both internal and external knowledge effectively.

5 Experiments

In this section, we detail the experimental setup and then report the main findings of our results.

LLMs	Natural Questions				TriviaQA				WebQuestions				SQuAD			
	JAcc	Hit	EM	F1	JAcc	Hit	EM	F1	JAcc	Hit	EM	F1	JAcc	Hit	EM	F1
Direct Retrieval-Augmented QA																
Llama2$_{7B}$	-	-	30.47	41.39	-	-	53.92	62.70	-	-	22.79	38.29	-	-	21.09	31.67
Llama2$_{13B}$	-	-	33.43	44.49	-	-	60.06	67.02	-	-	25.59	38.71	-	-	24.43	33.49
Mistral$_{7B}$	-	-	10.83	31.77	-	-	44.59	62.55	-	-	8.71	30.79	-	-	13.78	34.25
Baichuan2$_{7B}$	-	-	33.49	45.61	-	-	61.17	69.98	-	-	23.87	40.78	-	-	26.55	38.97
ChatGLM3$_{6B}$	-	-	13.27	20.48	-	-	24.57	33.76	-	-	5.61	18.38	-	-	8.31	15.98
Judge-then-Generate (4-shot)																
Llama2$_{7B}$	50.25	-	30.53	42.57	62.91	-	53.27	63.52	50.64	-	21.01	38.29	37.53	-	21.83	33.45
Llama2$_{13B}$	50.25	-	30.00	42.36	62.91	-	56.06	66.58	50.64	-	20.18	38.13	37.53	-	22.75	34.77
Mistral$_{7B}$	64.46	-	19.11	32.80	73.69	-	48.31	59.87	59.94	-	13.63	30.76	61.45	-	15.98	28.28
Baichuan2$_{7B}$	62.74	-	27.42	39.72	71.59	-	52.07	62.27	57.33	-	18.90	36.13	57.35	-	19.24	30.92
ChatGLM3$_{6B}$	52.74	-	24.65	32.67	63.79	-	46.57	54.23	51.57	-	20.37	34.60	58.70	-	18.71	25.90
Rank-then-Generate (4-shot)																
Llama2$_{7B}$	25.04	25.04	21.50	32.17	43.36	43.36	44.89	54.20	28.84	28.84	15.40	32.35	16.84	16.84	11.56	21.29
Llama2$_{13B}$	27.70	27.70	20.66	30.83	43.79	43.79	45.74	54.84	32.63	32.63	17.91	33.94	14.63	14.63	9.57	18.85
Mistral$_{7B}$	52.27	18.50	23.13	36.70	58.48	34.93	52.28	63.64	55.91	28.20	15.45	33.04	57.09	12.00	20.26	34.49
Baichuan2$_{7B}$	43.46	27.73	28.31	40.39	51.84	40.52	55.71	66.24	41.19	26.82	19.29	37.28	41.46	17.75	20.69	33.93
ChatGLM3$_{6B}$	54.38	33.27	20.03	26.72	49.77	43.42	39.87	46.99	47.29	34.45	16.73	29.57	46.45	19.67	17.08	24.22
Fine-tuned RALMs																
Self-RAG$_{7B}$†	19.81	51.11	41.52	-	35.69	64.47	47.59	-	25.69	47.98	31.40	-	10.73	38.73	25.47	-
Self-RAG$_{13B}$†	19.81	54.43	45.82	-	35.69	68.00	50.45	-	25.69	51.28	34.10	-	10.73	40.67	27.51	-
RobustLM$_{7B}$	49.75	-	39.28	48.67	37.09	-	61.76	70.14	49.36	-	36.27	48.68	62.47	-	20.12	29.80
REAR$_{7B\ w/\ pr}$	**74.85**	**68.25**	53.32	61.55	83.28	**77.34**	70.46	78.07	74.12	**57.93**	42.24	**52.54**	65.26	**60.83**	**41.88**	**53.47**
REAR$_{7B\ w/\ kc}$	74.27	67.67	**53.88**	**62.12**	**85.14**	77.22	**70.97**	**78.79**	**76.79**	57.86	**42.37**	52.49	**66.04**	60.78	41.73	52.81

Table 1: A comparison between REAR and baselines on NQ, TriviaQA, WQ and SQuAD datasets. JAcc, Hit, EM and F1 are short for judgment accuracy, Hit@1 (Recall@1), exact match scores and F1 scores. The best and second-best results are in **bold** and underlined fonts respectively. Self-RAG† is evaluated using accuracy (Acc) instead of EM, which is a less strict metric that measures whether the responses contain the answers. The last two lines are our REAR with different routing strategies: "w/ pr" means the path-reliability strategy, and "w/ kc" means the knowledge-consistency strategy.

5.1 Experimental Setup

Datasets. We extensively test on four open-domain QA datasets: Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016), and WebQuestions (Berant et al., 2013). We follow the training/validation/test split in prior work (Karpukhin et al., 2020) and report the results on the test set.

Baselines. We consider the following two lines of baselines for comparison.

(1) Retrieval augmentation based prompt methods: we design different prompting strategies based on open-source LLMs (without tuning tailored to RAG tasks) to support RAG, including

• *Direct Retrieval-Augmented QA*: We concatenate the top 10 retrieved documents as a single reference document for RAG. To enhance EM metric accuracy, we further incorporate several answer examples within the prompts, as illustrated in Fig. 4.

• *Judge-then-generate*: We following the approach of the prompting strategy (Yoran et al., 2023). The LLMs are required to determine document relevance before generating responses. It employs 4-shot demonstrations (Fig. 5), and provides the top 1 retrieved document.

• *Rank-then-generate*: We first task the LLMs with ranking the documents they are most inclined to deem "relevant" from the top 10 retrieved documents, by assessing the perplexity of their relevance judgments. Subsequently, the document selected is utilized by the LLMs for RAG.

For open-source LLMs, we consider Llama2-Chat (Touvron et al., 2023), Mistral-Chat (Jiang et al., 2023), Baichuan2-Chat (Yang et al., 2023), and ChatGLM3 (Du et al., 2022).

(2) Specially designed RAG methods: we also consider RobustLM (Yoran et al., 2023) and Self-RAG (Asai et al., 2023) as baselines, which have been specially optimized for the RAG tasks. To ensure a fair comparison, we fine-tune these two models with our training set. Furthermore, the two frameworks above are evaluated with the same set of retrieved documents as used for REAR.

Metrics. We employ four metrics to evaluate the model's capability of relevance assessment and QA accuracy. **JAcc**, which denotes judgment accuracy, measures the model's accuracy in assessing if the documents are relevant. Consistent with prior research (Khattab and Zaharia, 2020), documents are labeled as "*relevant*" if they contain the answers. **Hit** is the short form of "Hit@1", which evaluates if the document referenced for the model's final answer generation is relevant. Exact Match (**EM**) (Lee et al., 2019) calculates whether responses exactly match the gold truth answers. **F1** calculates the precision-recall overlap of predicted and true answers.

Implementation Details. To implement our REAR approach, we mix the four open-domain QA datasets to fine-tune Llama2-Base$_{7B}$ (Touvron et al., 2023). All retrieval documents are sourced from the top 10 documents retrieved by dense retrievers (detailed in Appendix C). More training and inference details are shown in Appendix D.

5.2 Main Results

Table 1 shows the results of REAR and baselines on four open-domain QA tasks.

First, our REAR approach surpasses all the other baselines in coarse-grained relevance binary discrimination (higher JAcc), and fine-grained document relevance comparison (higher Hit), achieving the best generation performance (higher EM and F1 scores) across the four datasets.

Second, REAR achieves superior results than RobustLM which employs the same training data as ours. This highlights the strengths of our architecture and training method. Furthermore, both RobustLM and REAR outperform Self-RAG, indicating that our data construction strategy is a crucial factor in enhancing RAG performance.

Third, while LLM-based baselines demonstrate a natural proficiency in discerning document relevance under few-shot scenarios (as seen in the JAcc metrics for *Judge-then-Generate* and *Rank-then-Generate*), they fall short in the nuanced comparison of relevance (noted in the Hit metrics for *Rank-then-Generate*). This limitation hampers their ability to effectively utilize multiple documents, highlighting an important scenario where our REAR architecture and training approaches excel.

Finally, LLMs like ChatGLM3 and Mistral exhibit better performance based on individual documents rather than given the top-10 documents as a reference on Natural Questions. This suggests that filtering irrelevant retrieved documents could be another viable option for enhancing RAG, rather than simply increasing document quantity.

Methods	Aspect	Hit@1	EM	F1
REAR	-	66.79	**53.13**	**61.84**
w/ Path-Reliability	Inf.	**67.48**	52.91	61.49
w/o Bi-granularity	Train.	66.54	51.88	59.91
w/o Noise-resistant	Train.	49.25	25.54	33.05
w/o Rank Head	Arch.	13.80	38.14	47.44

Table 2: Ablation study on our REAR. The "aspect" denotes the affected aspect. "Inf.", "Train" and "Arch." denotes inference, training and architecture respectively

5.3 Detailed Analysis

In this part, we further present the analysis of the ablation study and the impact of retrieved documents.

5.3.1 Ablation Study

We analyze how each of the proposed components affects final performance. Table 2 shows the performance of our default method and its four variants in three aspects, including the architecture, training algorithm and inference strategy.

(1) *w/o Rank Head*: the variant without the integration of the ranking module. We utilize language generation to assess relevance degrees instead of the rank head. The document is selected based on the probability of generating judgmental statements. There is a notable drop in the comparison accuracy (see Hit@1 metric), similar shortfall is also observed in Self-RAG (Table 1). This demonstrates the effectiveness of our architectural design, which not only minimizes interference between language generation and relevance discrimination, but also facilitates the incorporation of various loss functions.

(2) *w/o Bi-granularity*: the variant without bi-granularity fusion in relevance assessment training. We replace the bi-granularity loss with the coarse-grained loss function. The results indicate that the fine-grained relevance training could enhance the LLMs in relevance comparison among documents, and result in better performance.

(3) *w/o Noise-resistant*: the variant without noise-resistant training. We exclude the gold-noise data pairing, using the similar training construction approach of Self-RAG and RobustLM, with one document per query. We observe a notable decline,

LLM	Settings	Rel Doc (EM/Acc)	Irr Doc (EM/Acc)	Overall (EM/Acc)
Llama2_{7B}	4-shot	54.41	6.40	30.53
Llama2_{13B}	4-shot	53.36	6.40	30.00
Mistral_{7B}	4-shot	36.05	2.00	19.11
Baichuan2_{7B}	4-shot	48.68	5.96	27.42
ChatGLM3_{6B}	4-shot	46.97	2.12	24.65
Self-RAG_{7B}	fine-tuned	73.48	6.23	40.03
REAR_{7B}	fine-tuned	**73.84**	**20.09**	**46.79**

Table 3: Results of factual generation accuracy provided with top-1 retrieved documents on the test set of NQ. Categorized by performance when providing relevant (Rel) and irrelevant (Irr) documents.

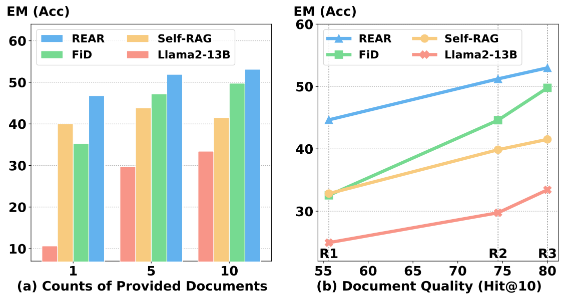

Figure 2: Results of RAG performance vary in overall document count and quality. The left one presents RAG performance with varying numbers of retrieved documents. The models are provided with the top 1, 5, and 10 document(s) retrieved by (Izacard and Grave, 2021a). The right one is the results of RAG with different retriever engines. R1, R2, and R3 represent BM25 (Robertson et al., 1995), Contriever-msmarco (Izacard et al., 2021), and the dense retriever presented by (Izacard and Grave, 2021a) respectively, with an overall retrieval capacity order as R1 $<$ R2 $<$ R3 (Table 5 of the Appendix) .

underscoring the effectiveness of noise-resistant training to enhance generation against irrelevant document interference.

(4) *w/ Path-Reliability*: using the path-reliability strategy instead of the knowledge consistency strategy. The path-reliability approach achieves higher Hit@1 rates, yet falls behind in EM and F1 scores compared to the knowledge-consistency strategy. The latter conducts a self-verification of outputs based on its generation ability, effectively integrating inherent knowledge in relevance assessment, which enhances the accuracy of RAG.

5.3.2 Impact of Retrieved Documents

In this part, we further analyze the impact of retrieved documents in both single-document and multi-document settings.

Single-Document Setting. We first examine the impact of external evidence in single document setting, where only the top first retrieved document is taken for reference. Table 3 shows the factual accuracy of different LLMs. We can see that both Self-RAG and REAR, after fine-tuning, perform well in relevant document utilization. However, REAR significantly outperforms other LLMs in generating accurate responses when the reference document is irrelevant, highlighting its robust resistance to interference from noisy documents.

Multi-Document Setting. In the second setting, we assume that multiple retrieved documents can be used for reference. Specially, we mainly examine the impact of the *total number* and *relevance degree* of reference documents. For this purpose, we vary the number of provided documents (Fig. 2(a)) and the retriever's capabilities (Fig. 2(b)). From Fig. 2(a), we can see that our REAR approach performs well when provided with a single document (*i.e.,* the top retrieved one), while base models without fine-tuning suffer from significant degradation in this case. Furthermore, as shown in Fig. 2(b), our approach is very robust to external retrievers of varied retrieval capacities. Especially, when equipped with the weakest retriever BM25, it yields a large improvement over the other baselines, which further demonstrates that our approach can effectively perceive the relevance of external evidence for more suitable utilization.

6 Conclusion

In this paper, we aimed to enhance the self-awareness of source relevance in RAG systems, and proposed **REAR**, a **RE**levance-**A**ware **R**etrieval-augmented approach for open-domain question answering (QA). For model architecture, we explicitly incorporated a specially designed rank head to precisely capture the relevance signals, and employed it to guide the utilization of external knowledge. For model training, we designed an improved training method with bi-granularity relevance fusion and noise-resistant training, which enhance the capacities of fine-grained relevance assessment and adaptive use of retrieved documents. Extensive experiments on four datasets demonstrate the effectiveness of REAR's relevance assessment and knowledge utilization.

As future work, we will extend the proposed approach REAR to dealing with more fine-grained source utilization (*e.g.,* passage or sentence level augmentation), and also consider applying REAR to other knowledge-intensive tasks.

Limitations

For LLMs, the challenge of being misled by irrelevant retrieved documents is a significant obstacle, underscoring the crucial need for enhancing LLMs' ability to adaptively utilize retrieved documents. In response to this issue, our work has concentrated on refining the architecture and training methods to bolster the effective use of retrieved documents by LLMs. We have implemented document-level relevance assessment and dynamic utilization strategies, significantly boosting the factual accuracy of generated content by LLMs. However, our current approach has not delved into guiding LLMs to focus more granularly on key sentences or tokens within the retrieved documents.

Moreover, the applicability of our methods across a broader spectrum of RAG tasks, such as those encompassed by the KILT benchmark, remains to be thoroughly evaluated. This gap presents a pivotal area for our future investigations.

References

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

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

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.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. *ACL 2020*, page 34.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv:2401.14887*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

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.

Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1437–1447.

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

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

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

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*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

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.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jingyuan Wang, Jian-Yun Nie, and Ji-Rong Wen. 2023b. The web can be your oyster for improving large language models. *arXiv preprint arXiv:2305.10998*.

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

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. *arXiv preprint arXiv:2106.00085*.

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*, pages 5835–5847, Online. Association for Computational Linguistics.

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

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

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. Pair: Leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *ArXiv*, abs/2304.09542.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

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

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2023. Take a step back: evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*.

Kun Zhou, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu, Rangan Majumder, Ji-Rong Wen, and Nan Duan. 2022. Simans: Simple ambiguous negatives sampling for dense text retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 548–559.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

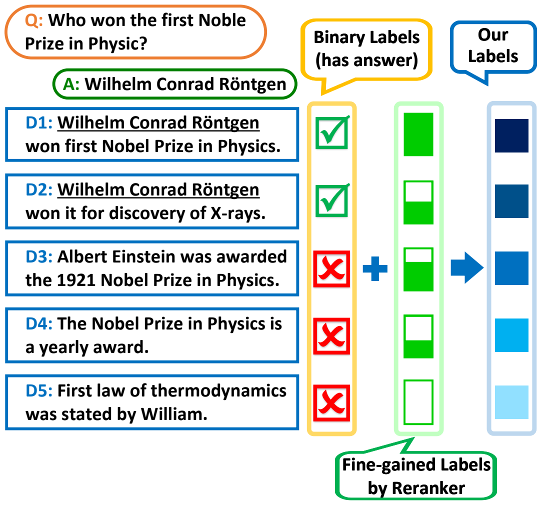

Figure 3: The illustration of different retrieved documents and different labeling metrics.

A Illustration of Fine-Gained Relevance Comparison and Details in Labeling

Given a query (Q) and the 5 retrieved documents (D1~D5) as shown in Fig. 3, traditional annotation methods in previous works typically use the presence of an answer within a document as the criterion to label it as positive or negative. In this illustration, both D1 and D2 meet this criterion. However, it's observable that while D1 directly answers Q, D2 merely states that Wilhelm was awarded for discovering X-rays, requiring some external knowledge for inference. Training models on simple binary classification does not differentiate the superiority of D1 over D2, potentially leading to inaccuracies in finer relevance judgment.

Inspired by previous work RankGPT (Sun et al., 2023) which attempts to train the rerankers by distilling GPT-4's ranking results. We opt for a more economical solution by annotating with small-scale, well-trained cross-encoder rerankers. Nonetheless, these models may still harbor the potential for misjudgment. To address this, we linearly combine the binary label with cross-encoder scores, ensuring that documents with answers are prioritized over those without, and within the same binary category, documents with higher cross-encoder scores are ranked above those with lower scores. To mitigate noise from rerankers, we disregard differences smaller than 0.1 in fine-gained relevance training.

B Details on Dataset.

We utilize four open-domain QA datasets, Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (WQ) (Berant et al., 2013) and SQuAD (Rajpurkar et al., 2016)).

• **NQ:** a dataset designed to support comprehensive QA systems. It includes questions sourced from actual Google search queries. The corresponding answers are text spans within Wikipedia articles, meticulously identified by human annotators.

• **TriviaQA:** a compilation of trivia questions paired with answers, both of which were initially extracted from online sources.

• **WQ:** constructed from questions proposed via the Google Suggest API, with answers being specific entities listed in Freebase.

• **SQuAD:** a dataset for evaluating reading comprehension, and also is used for training and testing open-domain QA engines.

Training Data Combination. We followed the same train, test, and dev set splitting approach as DPR (Karpukhin et al., 2020), In our study, we tailored the training dataset composition to optimize model performance across various question-answering tasks. Specifically, we integrated the complete datasets from Natural Questions (NQ), TriviaQA, and WebQuestions (WQ), applying a five-fold repetition to the WQ dataset to address its relatively smaller volume. For the SQuAD dataset, we opted to utilize only 2/9 of its training data, based on preliminary observations of its quicker convergence rates. Our detailed analysis (Section 5.3) explores the implications of employing a more straightforward strategy—merging the full training sets without repetition or selective sampling. Although this method simplifies dataset preparation, it yielded slightly lower performance compared to the aforementioned tailored approach.

Training Data Construction. We utilize task-specific dense retrievers to gather the top 100 retrieved documents from each dataset, creating a candidate document pool. Relevance scores for these documents are annotated using the reranker described in Appendix A. To avoid introducing false-positive and false-negative samples, we follow a strategy akin to RocketQA (Qu et al., 2021), excluding positive samples with scores below 0.1 and negative samples with scores above 0.9. The dataset is constructed using the two highest-scoring positive documents and six negative documents sampled based on Eq. 16, with parameters a, b, and k set to 15, 2, and 3, respectively.

C Details on Retrievers and Rerankers

Retrievers. For both training and inference, we employ advanced, task-specific retrievers to gather the necessary data. During the training phase, the RocketQAv2 (Ren et al., 2021b) retriever is utilized for the NQ dataset, and a dense retriever as described by (Izacard and Grave, 2021a) is used for the TriviaQA dataset. For the SQuAD and WQ datasets, we implement a strategy incorporating in-batch negatives and joint retriever-ranker training, starting from the Contriever-msmarco (Izacard et al., 2021) checkpoint. In the inference stage, to ensure a relatively fair comparison with FiD (Izacard and Grave, 2021b), we adopt the dense retriever method outlined by (Izacard and Grave, 2021a) to compile the top 10 documents. The retrievers we have trained are specifically applied during inference on the SQuAD and WQ datasets. The metrics of the retrieved documents for inference are shown in Table. 4.

Metrics	NQ	TriviaQA	WQ	SQuAD
Hit@1	50.25	62.91	50.64	37.53
Hit@10	80.00	81.78	75.89	68.51

Table 4: Retrievers we use for testing Hit rates (Recall rates) across datasets on the test sets.

Metric	**R1**	**R2**	**R3**
Hit@10	55.62	74.49	80.00
MRR@10	32.35	51.45	60.32

Table 5: Performance of three retrievers on the NQ test set. Hit@10 measures the percentage of correct answers within the top 10 results, indicating the precision of the retriever. MRR@10 (Mean Reciprocal Rank at 10) calculates the average of the reciprocal ranks of the first correct answer within the top 10 results, reflecting the effectiveness and rank of correct answers by the system. R1, R2 and R3 denote BM25 (Robertson et al., 1995), Contriever-msmarco (Izacard et al., 2021) and the dense retriever (Izacard and Grave, 2021a) trained by distilling attention scores of FiD reader (Izacard and Grave, 2021b)

Rerankers. For the NQ dataset, we utilize the RocketQAv2 reranker due to its publicly available checkpoint. Electra-base (Clark et al., 2019) serves as the backbone for training rerankers on the remaining three datasets, which are then employed

Knowledge:
{retrieved document 1}
{retrieved document 2}
......
{retrieved document 9}

Answer the following question with a very short phrase, such as "1998", "May 16th, 1931", or "James Bond", to meet the criteria of exact match datasets.

Question: {question}

Answer:

Figure 4: Instruction format for "direct RAG QA".

Given a passage and a query, predict whether the passage includes an answer to the query by producing either 'Yes' or 'No'. And then answer with the given passage if 'yes', or answer with your external knowledge if 'No'.

Passage: {positive document example 1}.
Query: {question 1}
Judge: Yes.
Answer: {answer 1}

Passage: {negative document example 2}.
Query: {question 2}
Judge: No.
Answer: {answer 2}

Other 2 examples...

Passage: {one of the retrieved document}.
Query: {question}
Judge: (calculate the difference in log perplexity for "Yes" and "No" and fill accordingly).
Answer:

Figure 5: Instruction format for "judge-then-generate" and "rank-then-genrate".

for annotating, filtering, and sampling the training data.

D Training and Inference Details

Training Settings Following the methodology of Self-RAG (Asai et al., 2023), our training employs a two-stage process. Initially, we focus on optimizing the LLM with coarse-grained loss as defined in Eq. 10. In the next stage, we apply joint optimization combining relevance assessment and generation adjustment, as specified in Eq. 14. The training utilizes a learning rate of 1e-6, a warm-up ratio of 0.03, a batch size of 64 and a cosine scheduler across two epochs. Our experiments leverage the computational power of 8 NVIDIA Tesla A100 GPUs, each with 40G of memory.

Inference settings. During the inference phase, we configure the threshold γ to 13 for binary discrimination of document relevance in Eq. 8 , and set λ to 9 to act as the knowledge-consistency scaler in Eq. 9. The generation process employs greedy generation.

E Details on Baselines

All open-source models employed in this study are obtained from huggingface, with each model's respective chat instruction format used for generation. No system instructions are utilized in any of the tests, and the greedy decoding strategy is applied for generation. The specific instruction formats used in our tests are illustrated in Fig. 4 and Fig. 5. In the "rank-then-generate" setup, we calculate the negative log perplexity $(-\log(ppl))$ for generating "yes" and "no", ranking them based on this metric to select the model that maximizes the probability of generating "yes" over "no".

F Computational Efficiency of Training and Inference

In this section, we discuss the computational efficiency of our model during both the training and inference phases. Our model introduces new modules and splits the inference process into two steps: relevance discrimination and answer generation. We employed various efficient fine-tuning and inference techniques, ensuring our model's computational demands during training and inference are comparable to those of the original LLM architecture.

Training Process. During training, we introduced negative examples in both bi-granularity fusion and noise-resistant training (Section 4.2), allowing the same instruction data to optimize two objectives simultaneously. This approach does not significantly increase computational overhead, akin to introducing a new word into the vocabulary to LLMs. Additionally, we adapted flash-attention (Dao, 2023;

Dao et al., 2022) techniques to accelerate training.

Inference Process. In the inference phase, we utilized the key-value (KV) cache (Vaswani et al., 2017) technology, making the computational cost of relevance discrimination equivalent to generating an extra token in the generation process. We also adapted vLLM (Kwon et al., 2023), leveraging PagedAttention for efficient inference.