# BIDER: Bridging Knowledge Inconsistency for Efficient Retrieval-Augmented LLMs via Key Supporting Evidence

**Jiajie Jin, Yutao Zhu, Yujia Zhou,** and **Zhicheng Dou***
Gaoling School of Artificial Intelligence, Renmin University of China
{jinjiajie,zhouyujia,dou}@ruc.edu.cn, yutaozhu94@gmail.com

## Abstract

Retrieval-augmented large language models (LLMs) have demonstrated efficacy in knowledge-intensive tasks such as open-domain QA, addressing inherent challenges in knowledge update and factual inadequacy. However, inconsistencies between retrieval knowledge and the necessary knowledge for LLMs, leading to a decline in LLM's answer quality. This paper introduces BIDER, an approach that refines retrieval documents into Key Supporting Evidence (KSE) through knowledge synthesis, supervised fine-tuning (SFT), and preference alignment. We train BIDER by learning from crafting KSE, while maximizing its output to align with LLM's information acquisition preferences through reinforcement learning. Evaluations across five datasets show BIDER boosts LLMs' answer quality by 7% while reducing input content length in retrieval documents by 80%, outperforming existing methods. The proposed KSE simulation effectively equips LLMs with essential information for accurate question answering.

## 1 Introduction

Large language models (LLMs) are currently developing rapidly and showing tremendous capabilities (OpenAI, 2023; Touvron et al., 2023). Nevertheless, they face challenges in knowledge updates and furnishing factual responses (Bang et al., 2023), especially in knowledge-intensive tasks like open-domain QA (Jiang et al., 2023b). To address these issues, retrieval-augmented generation (RAG) has emerged as a promising approach (Lewis et al., 2020; Guu et al., 2020; Tan et al., 2024). Retrieval-augmented methodologies serve to mitigate the drawbacks of LLMs by incorporating external knowledge, thereby enhancing the quality and reliability of generated answers (Izacard et al., 2023; Shi et al., 2023b; Press et al., 2023).

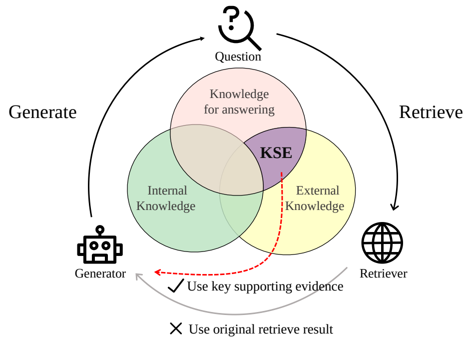


Figure 1: Key supporting evidence in RAG framework.

The standard RAG procedure involves retrieving pertinent documents related to a given question and subsequently inputting these documents as auxiliary information directly into the prompt. This strategic utilization enables the model to capitalize on its advanced text comprehension skills, facilitating the generation of precise and contextually appropriate answers.

However, retrieval-augmented LLMs are not always beneficial. Due to imperfections in the retrieval system and the inaccessibility of LLM's self-knowledge (Wang et al., 2023b), the retrieved documents provided to LLM are frequently lengthy and noisy, which can detrimentally affect generation quality (Petroni et al., 2020; Shi et al., 2023a).

Recognizing this decline, recent researches have made strides in optimizing retrieved documents. These efforts aim to mitigate noise in retrieved documents by employing sorting mechanisms to retain the most pertinent sentences (Xu et al., 2023; Arefeen et al., 2023), summarizing the retrieved text (Xu et al., 2023), and eliminating content that contributes minimally to the model's understanding (Li, 2023; Jiang et al., 2023a) or hinders effective generation (Yang et al., 2023).

While prior methods have shown progress in enhancing the quality of retrieved documents, they often rely excessively on feedback from the generator, overlooking the essential knowledge required

*Corresponding author



*Findings of the Association for Computational Linguistics ACL 2024*, pages 750–761

for addressing the questions themselves. This over-reliance on LLM's feedback is not only insufficient but also susceptible to the instability of LLM feedback, potentially resulting in the loss of crucial information and the retention of noisy elements. We argue that this limitation might stem from the neglect of knowledge inconsistency between the retrieved results and the knowledge truly required by the model for answering the question. We term this essential knowledge as Key Supporting Evidence (KSE). As shown in Figure 1, due to the imperfections of the retrieval system and the inaccessibility of LLM's self-knowledge (Wang et al., 2023b), retrieved results often contain numerous noise elements beyond key supporting evidence.

To address the aforementioned knowledge inconsistency issue, we propose BIDER(BrIDging knowledge inconsistency for efficient Retrieval-augmented LLMs), a method designed to refine retrieval documents into KSE. The overall training process of BIDER consists of three stages, integrating the strengths of both supervised and reinforcement learning, as shown in Figure 2. In the knowledge synthesizing stage, we employ a meticulous three-step process to synthesize authentic KSE. In the supervised fine-tuning stage, we construct a seq2seq model to learn the mapping from retrieved documents to KSE. Finally, in the preference alignment stage, we leverage reinforcement learning techniques to align the developed model with the preferences of the downstream LLM. This alignment ensures that the refined retrieval documents contain coherent and easily digestible key information, which is crucial for the LLM to generate accurate and informative responses.

We evaluate the effectiveness of our method on five datasets from three types of knowledge-intensive tasks, i.e., NQ, TQA, and HotpotQA for open-domain QA, WoW for dialogue generation, and FEVER for fact verification. Results show that our method achieves better generation performance while reducing the input information length by 80%, effectively condensing retrieved documents, and outperforming existing methods. We also validate the advantages of our proposed KSE data construction process and investigate the impact of the preference alignment stage on the final results. Furthermore, we validate the robustness of our approach under various text retrieval quality conditions.

The main contributions of this work are: (1) We propose a three-step knowledge synthesis method to generate oracle KSE. (2) We introduce a method to refine retrieval documents into KSE, thereby bridging knowledge inconsistencies between retrieval documents and the knowledge required by LLMs for answering. (3) We train the refiner model using supervised distillation and preference alignment techniques, efficiently enhancing RAG performance during inference by reducing input length and improving answer quality.

## 2 Related Work

### 2.1 RAG for LLMs

In knowledge-intensive tasks (Petroni et al., 2021), RAG (Lewis et al., 2020) has been introduced to enhance generative outcomes by incorporating external knowledge sources. In previous work, the retriever and generator are usually jointly trained end-to-end (Guu et al., 2020; Lewis et al., 2020; Borgeaud et al., 2022). With the advent of LLMs (OpenAI, 2023; Touvron et al., 2023; Zhu et al., 2024), most works now directly use them as generators due to their strong text comprehension ability, without the need for additional training (Jiang et al., 2023b; Yao et al., 2023; Shinn et al., 2023). While this approach demonstrates efficiency, it introduces new challenges, including susceptibility to interference from irrelevant content (Shi et al., 2023a; Bai et al., 2023; Mallen et al., 2022), insufficient attention to middle positions (Liu et al., 2023), and increased inference costs (Dettmers et al., 2022). Our method refines retrieved documents to eliminate noise, significantly reducing the input required for inference. By learning information retrieval preferences from LLM feedback, it provides the LLM with text that is more informative and easily captures relevant information, offering a substantial solution to the aforementioned issues.

### 2.2 Knowledge Refinement for RAG

Recent works leverage the capabilities of LLMs to identify pertinent information from various perspectives. Some approaches directly task the LLM with summarizing retrieval documents to identify pertinent information (Laskar et al., 2023; Chen et al., 2023; Gilbert et al., 2023; Xu et al., 2023). Moreover, certain methods employ smaller models to calculate perplexity as an importance indicator for filtering low-information text (Li, 2023; Jiang et al., 2023a). Xu et al. (2023) employ the LLM

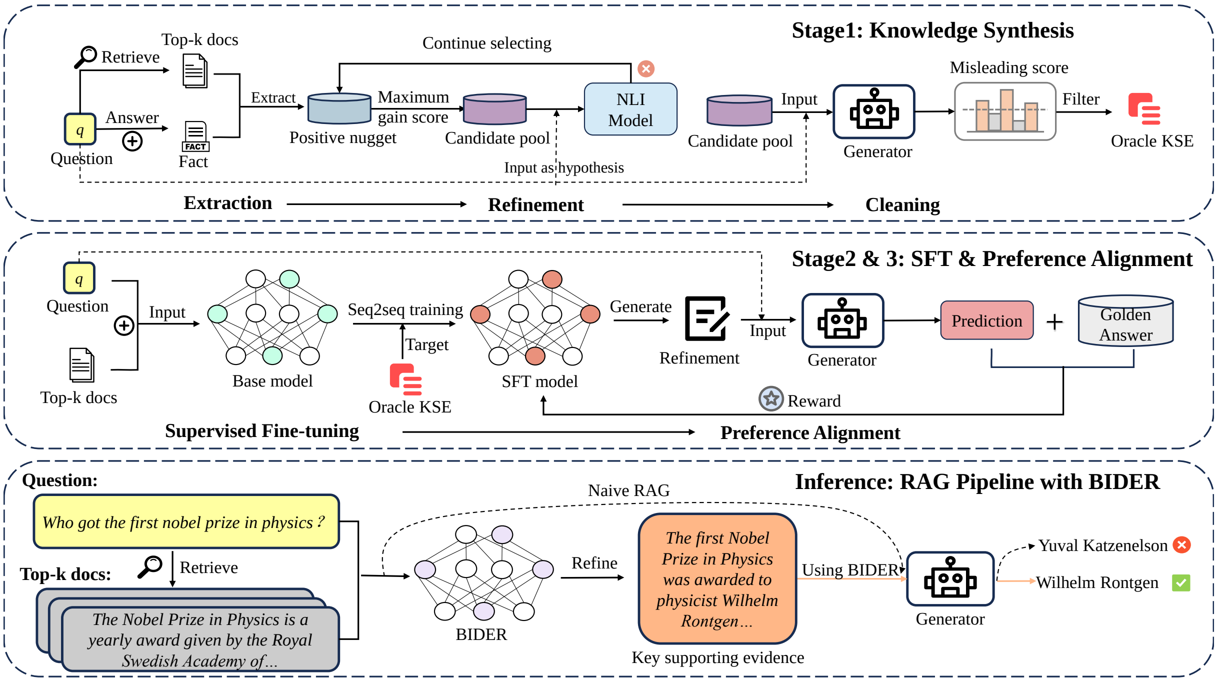


Figure 2: The overall architecture of BIDER. The first two lines represent the training process, which consists of three stages, and the last line represents the inference process of RAG with BIDER.

to assess the utility of each sentence in retrieval documents, using this information as labels to train a ranking model. Other works leverage LLM feedback for training; for instance, Arefeen et al. (2023) train a ranking model using reinforcement learning to retain top-ranked sentences, and Yang et al. (2023) design a reward mechanism to train a refinement model for retrieval documents. While these methods are effective, they are constrained by the instability of LLM feedback, providing limited guidance on specific information deemed valuable, which results in inefficient and suboptimal training outcomes. In contrast, our method employs well-designed key supporting evidence as a training objective, allowing the refiner to learn knowledge comprehensively before reinforcement learning, ensuring the provision of knowledge that better aligns with the LLM's needs. In summary, our approach offers two main advantages over these methods: (1) Broader practical application: our method not rely on annotated sentences or evidence, requiring only a golden answer to extract training targets; (2) Overall better performance: Our method optimizes both the SFT and RL phases, emphasizing the importance of their synergy, and can achieving better performance.

## 3 BIDER: a Knowledge Refiner for RAG

Our objective is to furnish the necessary knowledge for a generator, specifically the KSE as defined earlier, to answer a question. Since authentic KSE is unattainable, we employ a synthesize-and-learn paradigm. We design a method for synthesizing authentic KSE, training the refiner to learn the map from retrieval documents to constructed KSE, and adapting the model's information acquisition preferences based on the generator's feedback.

The overall framework of BIDER is illustrated in Figure 2. In this section, we first formulate the research problem (§3.1). Then, we introduce the details of three training stages of BIDER, including Knowledge Synthesis (§3.2), Supervised Distillation (§3.3), and Preference Alignment (§3.4).

### 3.1 Problem Formulation

In this problem, we assume that a document collection $\mathcal{C}$, a fixed retriever $\mathcal{R}$, and a fixed generator $\mathcal{G}$ are provided. For a given question $q$ and its corresponding golden answer $o$, we assume that $K$ documents are retrieved by retriever $\mathcal{R}$, denoted as $\mathcal{D}_q = \{d_i\}_{i=1}^{K}$. In the naive RAG framework, $\mathcal{D}_q$ is directly incorporated into the generator's input to obtain the output answer. We aim to find the optimal mapping function $\mathcal{F}^*$ for the retrieved documents $\mathcal{D}_q$, in order for the generator $\mathcal{G}$ to use $\mathcal{F}^*(\mathcal{D}_q)$ and achieve the best output results. We design BIDER to act as the mapping function, refining retrieval documents to make them more suitable for the input preferences of the generator.

### 3.2 Knowledge Synthesis Stage

We design a three-step method to gradually synthesize authentic KSE.

**(1) Nugget Extraction.** We initially narrow down the scope of knowledge helpful for answering by extracting nuggets from the retrieval documents. Here a nugget can be a sentence, a passage, or even a key phrase. In this paper, we use sentences as nuggets because using sentences already yields robust and consistent results. We will explore approaches with different nugget granularities in our future work. For each input question $q$ and its corresponding golden answer $o$, we first formulate them into a fact $f = \text{concat}(q, o)$ to ensure comprehensive semantic representation.[1] Then, we use $f$ as the query to perform sentence-level nugget retrieval in the retrieved documents $\mathcal{D}_q$ to remove noise and retain helpful sentences. In nugget retrieval, $\mathcal{D}_q$ is split into nuggets and transformed into vectors, while the query is vectorized. Based on the similarity between the query vector and nugget vectors, we obtain a positive nugget set $\mathcal{S}$ including retrieved top $K$ nuggets:

$$\mathcal{S} = \underset{s \in \mathcal{D}_q}{\text{TopK}}(\text{sim}(s, f)). \tag{1}$$

Here, $\text{sim}(\cdot, \cdot)$ represents the function for calculating semantic similarity by the E5 model (Wang et al., 2022), and $K$ is a hyperparameter. A larger $K$ can improve the recall of relevant information in $\mathcal{D}_q$, but it also raises the risk of including more irrelevant information.

**(2) Nugget Refinement.** While the extraction step effectively reduces noise in retrieved documents, there may be redundancy in $\mathcal{S}$. Therefore, we further design an iterative selection method to retain the minimal nugget subset necessary for answering the question.

Initially, we set up a candidate pool $\mathcal{P}$. In each round, we calculate a gain score for each nugget in $\mathcal{S}$, which represents the degree of assistance in answering the question. The gain score is defined as follows:

$$\kappa_i = \text{sim}(s_i, f) - \frac{1}{|\mathcal{P}|} \sum_{s_j \in \mathcal{P}} \text{sim}(s_i, s_j). \tag{2}$$

[1]For FEVER and HotpotQA where the answers have no actual semantic meaning, we use annotated evidence as a golden answer to ensure more accurate semantic information.

This takes into account the importance of the nugget itself as well as its duplication with the already-selected nuggets. Then, we select the sentence with the highest $\kappa_i$ from $\mathcal{S}$ and move it to the candidate pool $\mathcal{P}$. After moving, we use an NLI model to measure to what extent the candidate pool $\mathcal{P}$ can support answering $q$, i.e., yielding a support degree $\eta_k$ in $k$-th nugget selection.

The iterative selecting process will terminate in two cases: (1) when the support degree $\eta_k$ exceeds $\lambda_{\text{max}}$; (2)when the difference in support degree between two rounds, $\eta_k - \eta_{k-1}$, is less than $\lambda_{\text{min}}$, where $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$ are predefined thresholds. This aims to avoid introducing redundant information, especially in scenarios where retrieval documents fail to furnish adequate information, such as instances where the retriever's quality is subpar or when the posed question is challenging.

**(3) Nugget Cleaning.** The candidate pool $\mathcal{P}$ from the previous stage serves as the minimal subset of information necessary for answering. However, we have yet to consider the knowledge intrinsic to the generator itself, which encompasses information either known by LLM or detrimental to its generation. To mitigate conflicts arising from the disparity between external and internal knowledge, we conduct nugget cleaning in the candidate pool. In our experiments, we observe that the first nugget within the candidate pool is usually important. To avoid unintentionally removing vital information, we retain the first nugget directly and perform nugget cleaning for the left set.

For each nugget $s_i (i \geq 2)$ in $\mathcal{P}$, we assess its influence on the generator by determining whether it contributes to the model's output improvement when utilized as input. Specifically, we calculate the change in the log probability of generating the correct answer $o$ between the model's output before and after the inclusion of the nugget. This score for each nugget $s_i$ is denoted as

$$\tau_i' = \log \frac{\mathcal{G}(o|q \oplus s_i)}{\mathcal{G}(o|q)}, i \geq 2. \tag{3}$$

where $\mathcal{G}$ represents the generator.

Subsequently, we normalize all scores within the candidate pool to derive the final score $\tau_i$:

$$\tau_i = \frac{\tau_i'}{\sum_{j=2}^{|\mathcal{P}|} \tau_j'}, i \geq 2. \tag{4}$$

Nuggets with scores below $\lambda_{\text{lm}}$ are deemed unhelpful or potentially detrimental to the generator's response and are consequently excluded from the candidate pool. **The surviving nuggets in the filtered candidate pool represent the ultimate oracle KSE**, which correspond to the distillation results for each sample triplet $(q, o, \mathcal{D}_q)$.

### 3.3 Supervised Distillation Stage

In this stage, we aim to develop BIDER to acquire the ability to comprehend the relationship between retrieval documents and oracle KSE. This enhancement will enable BIDER to effectively refine its output during inference, particularly when provided only with the question.

A common approach is to consider this as a ranking task (Xu et al., 2023; Liu, 2019), using the nuggets extracted in the previous section as positive examples and other nuggets as negative examples for training the ranker. Although this method can relatively stably filter information, it is not able to effectively generate content that can adapt to the input of the generation model, as the refined content can only come from the original text.

We model the task as a seq2seq task, which is similar to the idea of pointer network (See et al., 2017; Gu et al., 2016). This method ensures the flexibility of refinement while enhancing the potential of the generation model in expression. Meanwhile, this serialization modeling approach makes it easier for the model to capture the generated sentences during generation. In Section 5.1, we will compare the two methods and demonstrate the effectiveness of our approach.

We use a pre-trained seq2seq model as the backbone model. For each sample triplet $(q, o, \mathcal{D}_q)$, the refiner model's input is the concatenation of the question and the original retrieval document: $q \oplus \mathcal{D}_q$. For ease of processing, we add separators between each document in $\mathcal{D}_q$ and merge them into one string. The target output of the model is the $\mathcal{P}$ extracted in the Knowledge Synthesis Stage, where each nugget is merged into one string in order. The training loss function of the model is the cross-entropy loss between the model output and the target output.

### 3.4 Preference Alignment Stage

Inspired by the RLHF technology (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022), we further enhance the adaptability of BIDER by incorporating feedback from a downstream LLM. Reinforcement learning enables the generative model to discern which nuggets are effective, thereby mitigating the possible risk of noise introduction in the knowledge synthesis phase. This complements our knowledge synthesis stage and helps our trained refiner better understand the context.

Specifically, we model the optimization problem of the model as a RL problem, with the objective to generate content that conforms to the LLM's information acquisition preferences without losing its original information capturing ability. The refiner model $\mathcal{M}$ to be optimized acts as a policy, where its action space encompasses all tokens in the vocabulary. We use the CLIP version of the PPO algorithm (Schulman et al., 2017) for optimization, which uses CLIP to control the magnitude of model updates. The loss function consists of three parts:

$$L_t^{\text{ALL}} = \mathrm{E}_t[L_t^{\text{CLIP}} - L_t^{\text{VF}} + L_t^{\text{BONUS}}]. \quad (5)$$

$L_t^{\text{CLIP}}$ is the primary objective function for optimizing the policy at step $t$, expressed as:

$$L_t^{\text{CLIP}} = \min(r_t \hat{A}_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t \quad (6)$$

$$r_t = \frac{\pi_\theta(y|x)}{\pi_{\text{old}}(y|x)}, \quad (7)$$

where $\epsilon$ is a hyperparameter to control the policy update magnitude, $r_t$ represents the conditional generation probability ratio between the new policy and the old policy, and $A_t$ denotes the estimated value of the advantage function at step $t$, calculated from Generalized Advantage Estimation (GAE) (Schulman et al., 2016):

$$\hat{A}_t = \sum_{l=0}^{T-t+1} (\gamma\lambda)^l \left(R_t + \gamma V(s_{t+1}) - V(s_t)\right),$$

where $\gamma$ and $\lambda$ are hyperparameters. $V$ represents the critic network used to estimate expected rewards, and $R_t$ indicates the reward at step $t$. $L_t^{\text{VF}}(\theta)$ is the squared error between the predicted reward and the actual reward output by the critic network, used to fit the critic network:

$$L_t^{\text{VF}}(\theta) = (V_\theta(s_t) - R_t)^2. \quad (8)$$

$L_t^{\text{BONUS}}(\theta)$ is an entropy bonus designed to ensure the model can explore sufficiently.

To calculate the above loss, we need a well-defined reward function $R_t$. Considering that the downstream LLM is highly sensitive to the overall information density and the position of key information, providing rewards before all sentences are generated could lead to inaccurate guidance. Thus, we design a segmented reward function:

$$R_t = \begin{cases} 0, & s_t \neq \langle \text{EOF} \rangle, \\ \text{F}_1(a_{\text{pred}}, o) - \text{F}_1(a_{\text{ori}}, o), & s_t = \langle \text{EOF} \rangle. \end{cases}$$

where $a_{\text{pred}}$ and $a_{\text{ori}}$ represent the answers generated by LLM based on the refiner result and original retrieval result respectively , $\langle \text{EOF} \rangle$ represents the end-of-sentence symbol. We generate answers from the LLM using the original document and refined results separately as references, and evaluate the quality of the refiner's distillation of the retrieved document by comparing the token-level $\text{F}_1$ scores of these two types of answers with the golden answer.

# 4 Experimental Setup

## 4.1 Datasets and Metrics

We experiment on five datasets of three knowledge-intensive tasks in the KILT benchmark (Petroni et al., 2021): (1) **Open-domain QA**, including NaturalQuestions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017),and HotpotQA (Yang et al., 2018); (2) **Dialog Generation**, including the Wizard of Wikipedia (WoW)(Dinan et al., 2019), where the generator is tasked with continuing the dialogue based on the preceding conversation history; (3) **Fact-checking**, including FEVER(Thorne et al., 2018) that classifies a given claim as "SUPPORTS" or "REFUTES".

We use Exact Match (EM) as the evaluation metric for NQ and TQA, use accuracy for FEVER, and use token-level $\text{F}_1$ (Jiang et al., 2023b) for HotpotQA and Wow. Evaluation is conducted on top 1000 samples in the test set of NQ, TQA, and HotpotQA, while on the development set of FEVER and WoW. Table 2 provides detailed sample sizes and the evaluation metrics used for each dataset.

## 4.2 Baselines

We compare with two types of baselines.

**Extractive Methods:** We employ three retrieval methods to extract sentences from retrieval documents, including BM25 (Xu et al., 2023), Sentence-BERT (Reimers and Gurevych, 2019), and LLM-Embedder (Zhang et al., 2023) which is trained with contrastive learning and feedback from LLM. To demonstrate the superiority of modeling the task as a seq2seq problem in the supervised distillation stage, we fine-tuned the bge-reranker-large (Xiao et al., 2023) for comparison.

**Abstractive Methods:** BART-Large is utilized for summarization (Lewis et al., 2019a), along with two state-of-the-art perplexity-based prompt refinement models: Selective Context (Li, 2023) and LongLLMLingua (Jiang et al., 2023a). Additionally, we include the RECOMP (Xu et al., 2023) method trained using distillation data from GPT-3.5 as a baseline.

## 4.3 Implementation Details

The size of the positive nugget set $K$ is set to 7. We utilize a T5-XXL model as the NLI model with the threshold $\lambda_{\text{max}}$ set to 0.5, $\lambda_{\text{min}}$ set to 0.01,$\lambda_{\text{lm}}$ set to 0.05.[2] We utilized BART-Large (Lewis et al., 2019b) as the base model for BIDER. During training, we utilize AdamW (Loshchilov and Hutter, 2019) as the optimizer with a learning rate of 5e-5, and a batch size of 32. In the preference alignment stage, $top_k$ is set to 10, and $top_p$ is set to 0.95. The training is implemented with HuggingFace Transformers (Wolf et al., 2020) and PFRL (Fujita et al., 2021). We use the December 2018 Wikipedia dump (Karpukhin et al., 2020; Lin et al., 2021) as retrieval corpus, BM25 as retriever, and SimLM as reranker (Wang et al., 2023a) for the top 100 documents returned by the retriever. LLAMA2-7B (Touvron et al., 2023) is utilized as the generator to provide answers.

For training for BGE-Reranker, we followed the fine-tuning procedure provided by the official FlagEmbedding repository.[3] The learning rate was set to 6e-5. We trained for five epochs with a batch size of 1 per device, and gradient accumulation steps were set to four. For the training data, only the top-ranked nuggets selected by us were labeled as positive examples, while the rest were marked as negative.

# 5 Experimental Results

## 5.1 Main Results

Table 1 reports the results of our approach alongside baseline methods on five knowledge-intensive datasets. It can be observed that our method outperforms the baseline on all datasets except FEVER, showcasing a notable performance advantage over other existing approaches, demonstrating around a 10% performance increase on all datasets. We also observe a relatively small performance gap on the

[2] https://huggingface.co/google/t5_xxl_true_nli_mixture

[3] https://github.com/FlagOpen/FlagEmbedding

| Methods | NQ | | TQA | | Fever | | HotPotQA | | Wow | | Avg | Avg tok |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EM | # tok | EM | # tok | Acc | # tok | F1 | # tok | F1 | # tok | | |
| ***Without refinement*** | | | | | | | | | | | | |
| Original Prompt | 0.356 | 725 | 0.480 | 787 | 0.517 | 805 | 0.376 | 770 | 0.086 | 710 | 0.363 | 759 |
| Zero-shot | 0.189 | 0 | 0.456 | 0 | 0.517 | 0 | 0.268 | 0 | 0.085 | 0 | 0.303 | 0 |
| ***Extractive refinement*** | | | | | | | | | | | | |
| BM25 | 0.295 | 163 | 0.479 | 181 | 0.520 | 193 | 0.356 | 186 | 0.085 | 177 | 0.347 | 180 |
| SBERT | 0.339 | 162 | <u>0.512</u> | 183 | 0.521 | 192 | 0.36 | 187 | 0.086 | 178 | 0.364 | 180 |
| LLM-Embedder | 0.357 | 161 | 0.503 | 179 | 0.522 | 192 | 0.352 | 186 | <u>0.118</u> | 179 | 0.370 | 179 |
| Bge-Reranker(top 5)* | 0.380 | 164 | 0.504 | 181 | 0.522 | 194 | <u>0.384</u> | 186 | 0.117 | 180 | <u>0.381</u> | 181 |
| Bge-Reranker (top 3)* | 0.362 | 81 | 0.491 | 79 | 0.516 | 94 | 0.342 | 124 | 0.114 | 88 | 0.365 | 93 |
| ***Abstractive refinement*** | | | | | | | | | | | | |
| BART-Summarizer | 0.326 | 185 | 0.507 | 204 | 0.518 | 215 | 0.369 | 254 | 0.085 | 194 | 0.361 | 210 |
| Selective-Context | 0.263 | 203 | 0.439 | 225 | 0.522 | 236 | 0.332 | 234 | 0.081 | 220 | 0.327 | 224 |
| LongLLMLingua | 0.221 | 251 | 0.433 | 175 | **0.551** | 111 | 0.302 | 222 | 0.077 | 124 | 0.317 | 177 |
| RECOMP | <u>0.397</u> | 48 | 0.497 | 60 | - | - | 0.356 | 90 | - | - | - | - |
| BIDER(ours) | **0.403** | 77 | **0.523** | 98 | <u>0.524</u> | 93 | **0.386** | 113 | **0.122** | 69 | **0.390** | 90 |

Table 1: Evaluation results on five knowledge-intensive datasets. The best results are in **bold** and second best results are <u>underlined</u>. The method marked with * have undergone additional training. For Bge-Reranker, we test two settings: extracting the top 5 and top 3 sentences to control for different token counts. For recomp, we only report the results on the three datasets it was trained on.

| Dataset | Task | Metric | # Train / #Test |
|---|---|---|---|
| NQ | Open-domain QA | EM | 79.1k / 2.6k |
| TQA | Open-domain QA | EM | 78.7k / 11.3k |
| HoPo | Open-domain QA | F1 | 88.8k / 5.6k |
| WoW | Dialogue | F1 | 63.7k / 3.0k |
| FEVER | Fact checking | Acc | 104.9k / 10.4k |

Table 2: Statistics and task metrics for five datasets.

FEVER dataset, indicating a potential weakness in the model's ability to leverage retrieval documents for text-based verification tasks. Compared to the original prompt, our method refines the retrieval documents to 20% of their original length, achieving an average improvement of approximately 8%. Notably, on the WoW dataset, the improvement approaches nearly 40%.

**Comparison with extractive methods.** The overall performance of extractive methods is quite satisfactory. Fine-tuning bge-reranker with our KSE extraction yielded the best results, indicating the effectiveness of our extracted KSE. However, there still exists a discernible gap between this approach and ours, possibly highlighting the influence of the preference alignment stage and model structure.

**Comparison with abstractive methods.** Abstractive refinement methods like Selective-Context and LongLLMLingua show a significant performance gap compared to our approach, particularly in QA tasks. This may result from their reliance on perplexity-based computations, posing a risk of losing essential entity information crucial for answering questions during refinement. In contrast, our method minimizes the risk of token-level information loss by employing sentence-level processing in data construction. Compared to RECOMP, our model demonstrates better performance, albeit with higher token usage. It seems that models trained on data distilled from GPT3.5 can be quite effective, but come with substantial annotation costs. In contrast, our method shows better overall results with considerably lower training costs, relying merely on locally deployed smaller models for target extraction.

### 5.2 Evaluation on Knowledge Synthesis Stage

To explore the necessity and effectiveness of the three steps in the knowledge synthesis stage, we use the results of each step as reference inputs for generating answers. For a comprehensive comparison, we incorporate results from the extraction based on the similarity between golden evidence and sentences in the retrieved documents.

As illustrated in Figure 3, with the further refinement of the retrieved text in the knowledge synthesis stage, the length of the input to the generator significantly decreases. However, there is a notable improvement in the quality of the LLM's

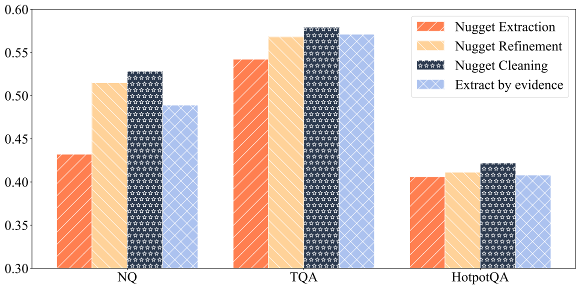


Figure 3: Performance of generator responses with different reference contents. 'Nugget Extraction', 'Nugget Refinement', and 'Nugget Cleaning' correspond to the two intermediate products and the final output in knowledge synthesis stage, respectively.'Extract by evidence' involves extracting the top 3 sentences based on the similarity between the golden target (answer or evidence) and sentences in the retrieved documents.

| Method | NQ | | TQA | |
|---|---|---|---|---|
| | EM | # tok | EM | # tok |
| BIDER | 0.403 | 77 | 0.523 | 98 |
| w/o preference alig. | 0.373 | 94 | 0.518 | 85 |
| w/o knowledge syn. | 0.340 | 118 | 0.504 | 133 |
| Original retrieval results | 0.356 | 725 | 0.480 | 787 |

Table 3: Ablation study on NQ and TQA.

responses. This observation indicates the effectiveness of our approach in reducing noise in the retrieved text, providing the generator with more easily exploitable information. Simultaneously, it is observed that directly using golden evidence as the target for information extraction results in an inferior performance. Overall, the effectiveness is somewhat lower compared to our second step. This suggests that relying solely on the relationship between the text and the question/answer for data extraction is insufficient, and it's necessary to consider the knowledge of the model itself when constructing the data.

### 5.3 Ablation Study

To assess the impact of BIDER's key components, we performed ablation experiments on NQ and TQA. Two variants were introduced for study: i) *BIDER w/o preference alignment*, using models without reinforcement learning, and ii) *BIDER w/o knowledge synthesis*, replacing knowledge synthesis method with a naive sentence-level retrieval method as the training target for SFT.

Table 3 displays the results, emphasizing a decline in performance when either component is removed. This underscores the indispensability of both components. Particularly, the impact on performance due to the absence of the knowledge synthesis method is more significant than that of the preference alignment part. This implies that the construction of the training target in the initial phase is more crucial than preference alignment. Hence, emphasizing the construction of training data in the first phase should be a priority, rather than relying solely on LLM feedback for learning.

| Dataset | Align | EM | % Gold in Out. | Avg Gold Pos. |
|---|---|---|---|---|
| NQ | × | 0.373 | 48.1 | 1.33 |
| | ✓ | 0.403 | 51.1 | 1.19 |
| TQA | × | 0.518 | 56.4 | 1.14 |
| | ✓ | 0.523 | 61.1 | 1.10 |

Table 4: The comparison of refiner results with and without preference alignment on NQ and TQA. 'Avg Gold Pos.' represents the average position of sentences containing the golden answer (calculated only on samples that include the golden answer).

| Method | BM25 | | BM25+SimLM | |
|---|---|---|---|---|
| | EM | # tok | EM | # tok |
| Original Prompt | 0.257 | 716 | 0.356 | 725 |
| LLM-Embedder | 0.278 | 165 | 0.357 | 161 |
| BAET-Summarizer | 0.269 | 183 | 0.326 | 185 |
| BIDER(ours) | 0.325 | 80 | 0.403 | 77 |

Table 5: Experiments with different retrievers on NQ.

### 5.4 Impact of Preference Alignment

To further investigate the impact of preference alignment, we analyzed model output before and after this stage, specifically focusing on effective information content and its optimal sequence. We measured the proportion of golden answers in the generated results and their average position on a sentence level for models trained through supervised learning and those additionally trained with preference alignment.

Table 4 presents the results, indicating that after preference alignment training, the proportion of golden answers in the model output increased by 3%-4%, and their position in the output text moved closer to the beginning. This improvement suggests a dual effect: an augmentation in information content and a repositioning of crucial information towards the text's forefront.

### 5.5 Impact of Retrieval Quality

We directly utilize top 5 retrieval documents from BM25(without reranker) to demonstrate generalizability on weaker retrievers. As depicted in Table 5,

| Method | BIDER | Generator | Total |
|---|---|---|---|
| End-to-End w/o BIDER | \ | 1.33 | 1.33 |
| End-to-End w/ BIDER | 0.10 | 1.08 | 1.18 |

Table 6: Inference latency (seconds/query) on NQ.

our approach performs well under different quality retrievers, surpassing other methods. And it can be observed that our method brings more improvements when the retriever quality is worse, indicating the effectiveness of our refinement method.

### 5.6 Inference Latency

Table 6 shows the inference latency of various components within the system on a V100-32G GPU. It is observed that the time required for text refinement using BIDER is notably short, facilitating effective support for applications in the RAG scenario. Additionally, as the refined input to the generator is shorter, the time taken by the generator to produce responses has also decreased. Consequently, there is a 10% enhancement in the overall end-to-end speed.

## 6 Conclusion

We present BIDER, a method to refine retrieved documents into KSE, addressing inconsistencies between retrieved results and the knowledge needed by the generator. We designed a three-step process to synthesize authentic key supporting evidence to enhance the effectiveness of supervised learning, while utilizing LLM's feedback for further alignment. Through a well-structured training process, BIDER effectively provides the generator with the necessary information to answer questions based on the original retrieval text, achieving a significant improvement in answer quality while reducing input length by 80%.

## Limitations

Our approach has some limitations. It performs less effectively in complex datasets like HotpotQA compared with NQ and TQA, suggesting that additional factors need to be considered for complex tasks. Also, our method requires separate training for each dataset and generator, limiting its use across different tasks and generators. Lastly, our datasets are based solely on Wikipedia, while real-world RAG applications involve diverse sources with varied writing styles. Optimizing for this diversity may require further refinement.


## Acknowledgments

This work was supported by Beijing Natural Science Foundation No. L233008, National Natural Science Foundation of China No. 62272467, the fund for building world-class universities (disciplines) of Renmin University of China, and Public Computing Cloud, Renmin University of China. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.


## References


Md. Adnan Arefeen, Biplob Debnath, and Srimat Chakradhar. 2023. Leancontext: Cost-efficient domain-specific question answering using llms. *CoRR*, abs/2309.00841.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *CoRR*, abs/2308.14508.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 675–718. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *CoRR*, abs/2310.05029.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in*

*Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yasuhiro Fujita, Prabhat Nagarajan, Toshiki Kataoka, and Takahiro Ishikawa. 2021. Chainerrl: A deep reinforcement learning library. *J. Mach. Learn. Res.*, 22:77:1–77:14.

Henry Gilbert, Michael Sandborn, Douglas C. Schmidt, Jesse Spencer-Smith, and Jules White. 2023. Semantic compression with large language models. In *Tenth International Conference on Social Networks Analysis, Management and Security, SNAMS 2023, Abu Dhabi, United Arab Emirates, November 21-24, 2023*, pages 1–8. IEEE.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *CoRR*, abs/2310.06839.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel 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, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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

Md. Tahmid Rahman Laskar, Mizanur Rahman, Israt Jahan, Enamul Hoque, and Jimmy Huang. 2023. Cqsumdp: A chatgpt-annotated resource for query-focused abstractive summarization based on debatepedia. *CoRR*, abs/2305.06147.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019b. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering. *CoRR*, abs/2304.12102.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Frassetto Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2356–2362. ACM.

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. *CoRR*, abs/2307.03172.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *CoRR*, abs/2212.10511.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *CoRR*, abs/2005.04611.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. REPLUG: retrieval-augmented black-box language models. *CoRR*, abs/2301.12652.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for llms. *CoRR*, abs/2402.12052.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023a. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023b. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. RECOMP: improving retrieval-augmented lms with compression and selective augmentation. *CoRR*, abs/2310.04408.

Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5364–5375, Singapore. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *CoRR*, abs/2310.07554.

Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zhicheng Dou, Zheng Liu, and Ji-Rong Wen. 2024. INTERS: unlocking the power of large language models in search with instruction tuning. *CoRR*, abs/2401.06532.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.