Prompting is not Enough: Exploring Knowledge Integration and Controllable Generation on Large Language Models

Anonymous Author(s)

No Institute Given

Abstract. Open-domain question answering (OpenQA) represents a cornerstone of research in natural language processing (NLP), primarily focused on extracting knowledge-based answers from unstructured textual data. With the rapid advancements in Large Language Models (LLMs), LLM-based OpenQA methods have made significant strides in comparison to traditional methods, which have reaped the benefits of emergent understanding and answering capabilities enabled by massive parameters. However, most of these methods encounter two critical challenges: how to integrate knowledge into LLMs effectively and how to adaptively generate results with specific answer formats for various task situations. To address these challenges, we propose a novel framework named **GenKI**, which aims to improve the OpenQA performance by exploring **K**nowledge **I**ntegration and controllable **Gen**eration on LLMs simultaneously. Specifically, we first train a dense passage retrieval model to retrieve associated knowledge from a given knowledge base. Subsequently, we introduce a novel knowledge integration model that incorporates the retrieval knowledge into instructions during fine-tuning to intensify the model. Furthermore, to enable controllable generation in LLMs, we leverage a certain fine-tuned LLM and an ensemble based on text consistency incorporating all coherence, fluency, and answer format assurance. Finally, extensive experiments conducted on the TriviaQA, MSMARCO, and CMRC2018 datasets, featuring diverse answer formats, have demonstrated the effectiveness of GenKI with comparison of state-of-the-art baselines. Moreover, ablation studies have disclosed a linear relationship between the frequency of retrieved knowledge and the model's ability to recall knowledge accurately against the ground truth. Tests focusing on the out-of-domain scenario and knowledge base independence scenario have further affirmed the robustness and controllable capability of GenKI. Our anonymous code is available at https://anonymous.4open.science/r/GenKI-C0B8[1]

Keywords: Open Domain Question Answering · Large Language Model · Knowledge Integration · Controllable Generation.

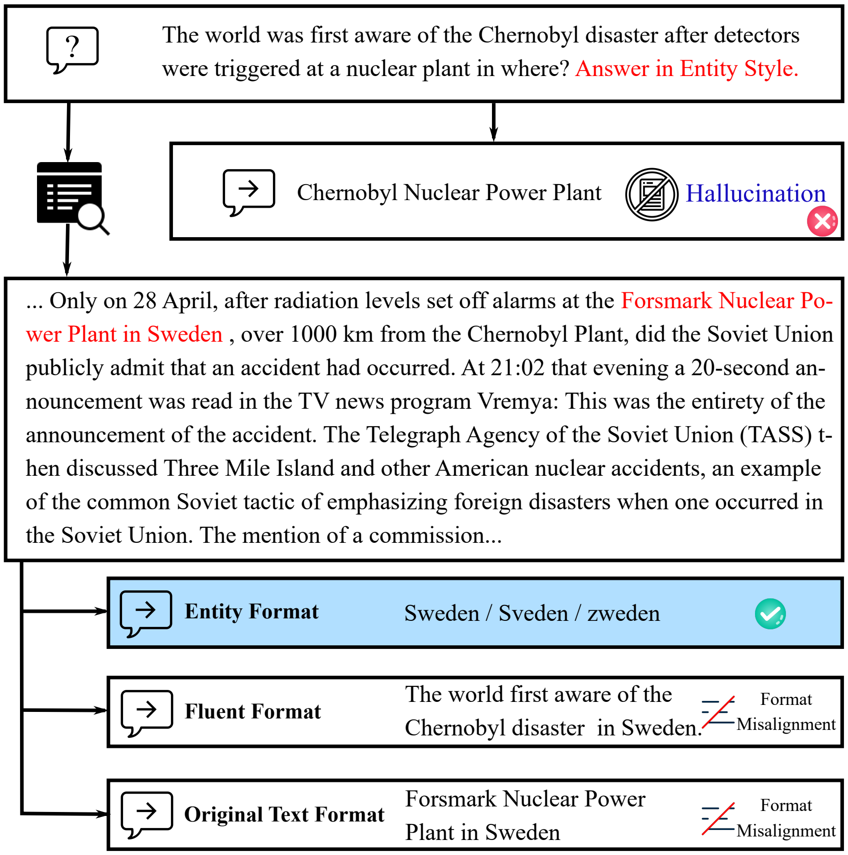
[1] This is a paper authored by a student as the first author.

1 Introduction

Open-domain question answering (OpenQA) has garnered significant attention in the Natural Language Processing (NLP) community, as it offers enhanced user-friendliness and efficiency compared to conventional search engines. OpenQA seeks to respond to questions utilizing auxiliary knowledge sources [57], necessitating models that possess both knowledge capability and comprehension abilities. Conventional methods usually entail a retriever-reader framework [12]. The evolution of retriever models has progressed from BM25 [41] and TF-IDF [2] to dense-vector-based approaches, such as DPR [24] and SEAL [5]. Concurrently, reader models have diversified, spanning from extractive readers like DPR-reader [11] to generative readers, such as FiD [20] and RAG [28], targeting span and free answering tasks respectively. Recently, the rapid development of Large Language Models (LLMs) has motivated researchers to incorporate them into OpenQA, driven by the models' advanced abilities in natural language reasoning. For instance, ICL [29] contains to use certain prompts towards better integration of external knowledge with in-context learning, while Replug [43] enhances an LLM with retrieved knowledge as prompts.

However, when utilizing language models for OpenQA, two notable error phenomena persist, as depicted in Figure 1. These errors include: (1) **Knowledge Deficiency**. When prompted to identify "*which nuclear plant's detectors first alerted the world to the Chornobyl disaster*", the LLM fabricated an incorrect response. This issue stems from the limited memorization capacity of LLMs. Specifically, LLMs not only face challenges in retaining less frequent knowledge [23], but also tend to produce hallucinations [44] when the requisite knowledge is absent from their pretraining data. (2) **Answer Format Alignment**. Various OpenQA datasets demonstrate unique answer formats, exemplified by the three markedly different formats depicted in Figure 1. However, even when tasked to produce answers in a specific format, the outputs of LLMs often diverge from these expected formats. This discrepancy is largely due to the distributional biases present in the pre-training corpora, as discussed in [23].

Presently, to tackle challenge (1), PaLM 540B [8] enlarges the model scale to accommodate a greater wealth of knowledge, which, however, leads to unacceptable computational complexity. So the ideology of RAG [17] integrating knowledge with prompts for efficiency has raised. After researchers recognized the significance of retrieving knowledge, they sought to enhance the quality of retrieved content. Typical examples are enabling large models to autonomously assess the quality of retrieved content for answering questions [40], or iteratively conducting multi-step retrievals to progressively enhance the quality of retrieved content and align with the preferences of LLM[3]. Nevertheless, integrating knowledge through prompt-based approaches treats LLMs merely as instruction interpreters, disregarding their potent knowledge storage capabilities [39]. Consequently, this approach yields unstable and suboptimal results. Toward challenge (2), InstructGPT [38] and FLAN [50] adopt a pre-training and instruction tuning paradigm and attempt to encapsulate all instructions along with the language understanding capabilities equipped with knowledge at the same time. Nevertheless, this paradigm will require the storage of massive real-world data, and the computational burden is quadratic with respect to the number of model parameters [18]. To address this issue, CoF-CoT [36] attempts to utilize a chain of instruction to control the output format. However, a distribution gap persists between the formatted output and knowledge. Consequently, the collaborative training procedure is prone to encountering contradictions [32].



The world was first aware of the Chernobyl disaster after detectors were triggered at a nuclear plant in where? Answer in Entity Style.

... Only on 28 April, after radiation levels set off alarms at the Forsmark Nuclear Power Plant in Sweden , over 1000 km from the Chernobyl Plant, did the Soviet Union publicly admit that an accident had occurred. At 21:02 that evening a 20-second announcement was read in the TV news program Vremya: This was the entirety of the announcement of the accident. The Telegraph Agency of the Soviet Union (TASS) then discussed Three Mile Island and other American nuclear accidents, an example of the common Soviet tactic of emphasizing foreign disasters when one occurred in the Soviet Union. The mention of a commission...

The world first aware of the Chernobyl disaster in Sweden.

Fig. 1: An illustration of hallucination encountered when employing LLMs in OpenQA task, along with the variations in answer formats across different OpenQA datasets.

In general, although the aforementioned generator enhancement methods have achieved remarkable progress in the OpenQA field, current pre-training or fine-tuning-based methods only focus on either full knowledge adaption or applying LLM on both reasoning and answer alignment missions, considering two different kinds of task in one stage of tuning. This leads to distribution misalignment and limitations in sufficient knowledge integration and controllable generation performance in different task situations.

Towards these challenges, in this paper, we propose a novel framework named **GenKI**, to improve the OpenQA performance by exploring **K**nowledge **I**ntegration and controllable **Gen**eration on LLMs. We introduced a novel three-stage paradigm for retrieval, knowledge integration, and controllable generation instead of the two-stage paradigm in RAG, ensuring that the model focuses on either knowledge integration or controlled generation in each instance, to avoid the distributional differences brought by these two tasks. Specifically, in the knowledge integration model, rather than adopting on normal fine-tuning-based method, we propose an innovative method of LLM combining autoregressive training loss and supervised fine-tuning loss to involve the retrieval results. In comparison to the prior research on in-context learning using RAG or fine-tuning methods, which solely instruct the LLM, our model acquires new knowledge more stably by storing the retrieved domain knowledge within parameters rather than relying on prompts. In the controllable generation model, we first utilize a certain fine-tuned LLM to post-process the generated answer towards a certain format. Subsequently, we propose a novel ensemble methodology based on text consistency to incorporate both coherence and fluency from the reward model and faculty from the external selection, thereby ensuring adaptive alignment between the output of our model and the target format.

Extensive experiments on three benchmark datasets, including TriviaQA, MSMARCO, and CMRC2018, have demonstrated the effectiveness of the proposed framework. Additionally, our ablation study results reveal: (1) a linear relationship between the quality of retrieved results and the model's knowledge proficiency, and (2) the impact of different structures on our answers, with the tuning mechanism playing a key role in generating answers in a specific format, and the reward model excelling in choosing fluent sentences. The main contributions could be summarized as follows:

- We propose a novel problem on the phenomena of knowledge deficiency and answer format misalignment when applying LLMs to the OpenQA task.
- We introduce a novel three-stage paradigm of retrieval, knowledge integration, and controllable generation, avoiding the distributional differences of LLM in two-stage RAG, and propose a novel ensemble method based on text consistency, ensuring coherence and faculty of the generated answer.
- Extensive experiments on three benchmark datasets and analysis on OOD situation and independence knowledge base tests unequivocally affirm the distinguished controllable generation and robustness of our framework.
- Through ablation experimental results, we find the linear fitting relationship between the quality of retrieved results and the knowledge proficiency of the model and a way to judge whether the model is pre-trained on knowledge demanded, for future research on knowledge integration in LLMs.

2 Related Work

2.1 Open-domain Question Answering

OpenQA is a crucial task in NLP, which aims to answer information-seeking questions based on auxiliary knowledge source [22]. Traditionally, an OpenQA approach entails a retriever-reader framework, with the retriever used to locate relevant knowledge and the reader generating the response. Extensive research has delved into retriever techniques, encompassing a range from traditional ones such as BM25 [41] and TF-IDF [2] to dense-vector-based methods like DPR [24], as well as retrievers like SEAL [5], which leverages all n-grams in a passage as its identifiers. Since then Iterative retrievers such as GRAPHRETRIEVER [35] and GAR [33] have been raised for answering complex questions like those requiring multi-hop reasoning task [53]. Reader models exhibit a spectrum ranging from extractive readers to generative readers. Extractive readers develop from origin DPR-reader [11] to graph-based readers like GRAPHREADER [35] and readers ensembling methods like BERTserini [52] focus on span-based text answering [11]. In contrast, generative readers such as FiD [20] and RAG [28] focus on free-form text answering [4,22]. Certainly, given the impressive capabilities showcased by LLMs, developed after generative readers, traditional approaches have witnessed a shift in prominence, often assuming supplementary roles alongside these models. However, valuable methods can still be gleaned from these works. In our study, one such approach, namely Dense Passage Retrieval (DPR), was integrated as a retrieval plugin.

2.2 LLMs-based Open-domain Question Answering

Recent years have witnessed a surge in interest in LLMs for their remarkable capabilities across various NLP tasks like QA and NER [34]. Amidst this trend, handling OpenQA based on LLMs is emerging as a popular research direction. Research in this domain can be mainly divided into two categories: discriminative language models-based approaches and generative language models-based approaches. In the first type of approaches, researchers typically fine-tune BERT [13] or RoBERTa [30] to build a reader aligned to answering tasks [21,51,54]. Following the occurrence of generative language models, such as GPT [7], GLM [15], LLaMA [47], the researchers began adopting these models for OpenQA. This shift was driven by their proven capability to handle OpenQA without relying on retrievers [49]. Nevertheless, Self-Prompting [29] persists in trying to enhance LLM itself by in-context learning. There is also some research exploring the role of retrievers in enhancing the performance of generative large models on these tasks, such as REPLUG [43] and kNN-LM [56]. Afterward, researchers recognized the effectiveness of the paradigm using retrieved knowledge, i.e., RAG [17], and began to work on further refining this paradigm.

2.3 RAG-LLMs on Open-domain Question Answering

Advanced RAG [17] has made improvements to overcome the deficiencies of Naive RAG. The current emphasis of the Advanced RAG model is on enhancing the Retriever and Generator components. On the Retriever enhancement, LangChain [46] first proposed a modularized retrieval and reranking framework. RRR [31] and AAR [55] try to align Retriever with LLM using modularized or end-to-end structure. Finally, the LLaMA-index[2] proposed an end-to-end alignment and rerank structure to merge them. On the Generator enhancement, end-to-end frameworks [42] and Pre-training methods like PaLM [9] are proposed for pre-training LLMs adapting to downstream knowledge. Nevertheless, with the expanding scale of large language models, the viability of the pre-training method diminishes over time. So recently prompt-engineering [29] and supervised fine-tuning[38] are the technologies widely used. RFiD [48] finetunes the decoder to generate answers by relationships and features. GAG [1] finetunes the model for both answering and generating contextually rich documents tailored to the given question. Self-rag [3] further finetunes an LLM generates and reflects on retrieved passages and its own generations using special tokens. Our framework is also focusing on generator enhancement. As this framework is independent of the retrieval content, all the retrieval optimization solutions such as llama-index and GRAPHRETRIEVER mentioned above can be directly applied to this framework.

Although the aforementioned generator enhancement methods have achieved remarkable progress in the OpenQA field, current pre-training or fine-tuning-based methods only focus on either full knowledge adaption or applying LLM on both reasoning and answer alignment missions. This leads to limitations in sufficient knowledge integration and controllable generation performance in different task situations, which is also what this paper focuses on.

3 Preliminary

3.1 OpenQA Problem Definition

Given a domain knowledge base (e.g., Wikipedia) comprising a set of sentences $P = \{p_1, p_2, ..., p_{n_1}\}$, a specified answer format request F, and each query question q_i within $Q = \{q_1, q_2, ..., q_{n_2}\}$, OpenQA aims to train a model M that can extract relevant knowledge and generate the ideal answer $a_i = M(q_i, F, P)$ in response to each query, guided by the information in the knowledge base.

In authentic knowledge bases like Wikipedia, information is typically structured within web pages or documents. By segmenting it into sentence form, we can derive the set of sentences denoted as P. Take, for instance, the question q_i: "*Where was the initial awareness of the Chernobyl incident triggered?*". Given that the desired format F is Entity Style, the model is expected to leverage insights extracted from these sentences to generate an answer such that $a_i = M(q_i, F, P) =$"*Sweden*".

[2] https://www.llamaindex.ai

3.2 Instruct Tuning on LLMs

In the OpenQA pipeline with LLMs, we employ a fine-tuned LLM denoted as the model M for answer formulation. The fine-tuning process of the LLM facilitates its adaptation to the distribution and domain knowledge relevant to downstream tasks. This process essentially encompasses an equivalent final objective loss, resembling that of autoregressive training, outlined as follows:

$$max_{\Phi} \sum_{x_i, a_i \in T} \sum_{t=1}^{|a_i|} log(P_{\Phi}(a_{i,t}|x, a_{i,<t})), \tag{1}$$

where Φ represents the parameters of LLM to be optimized, T denotes the training set, x refers to the input context encompassing both an instruction and a query question, and $a_{i,t}$ is the t-th token of the generated answer word. For all tunings of our experiment, we adopted the LoRA [19] method of lightweight fine-tuning by reducing the required GPU memory consumption.

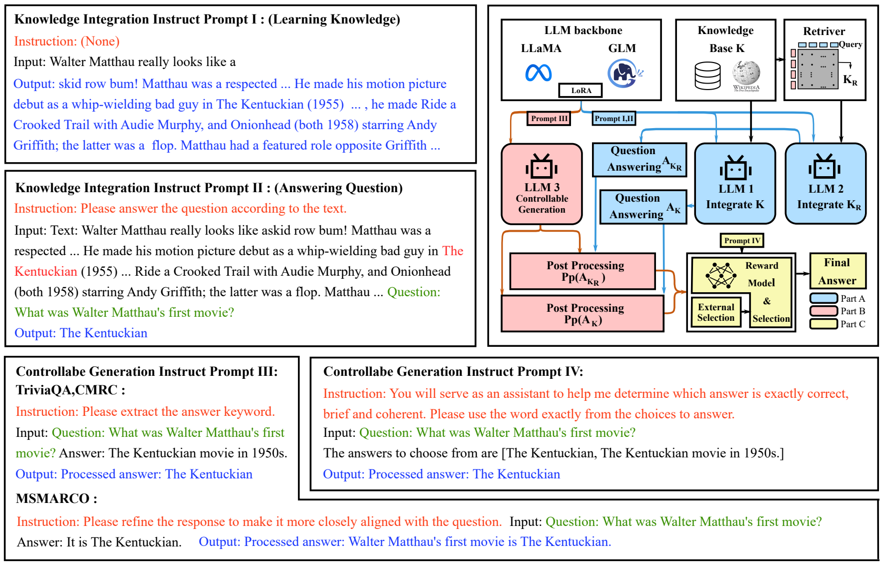
4 Methodology

To address the knowledge deficiency and uncontrollable generation issues in LLMs for OpenQA, our system combines three modules as illustrated in Figure 2. On the whole, we decompose the model $A = M(q, F, P)$ into $A = C(A_K, F) = C(K(q, P), F)$ is to ensure that the model focuses on either knowledge integration or controlled generation in each instance, to avoid the distributional differences brought by these two tasks. Specifically, we first train a **Knowledge Retriever** module $R(P, q) = P_R$ to retrieve knowledge according to the query supplying the model during knowledge integration steps. Subsequently, once the retrieved knowledge is synthesized, a **Knowledge Integration** module $K(P_R, q) = A_K$ (Part A in Figure 2) is deployed to generate answers by integrating the retrieved knowledge. Finally, a **Controllable Generation** module $C(A_K) = A$ (Part B in Figure 2) involves using a fine-tuned LLM on the target dataset format for post-processing the results and a Reward model to further ensure that the output of our method is controlled. We split $C(A_K) = Ensemble(Rm(A_K), Ext(A_K))$ (Part C in Figure 2) to gain formatted, coherent and factual answer. The function of each module will be introduced in detail in the following subsections.

4.1 Knowledge Retriever

Firstly, given vast knowledge repositories, we should prioritize enabling our model to learn from them. However, feeding all the sentences P from the knowledge base into the LLM may potentially result in the model's inability to grasp essential focal points. Therefore, we need a retriever $R(P, q) = P_R \subset P$ as the initial input into the next knowledge integration module K.

Specifically, we use a DPR to retrieve knowledge from the knowledge base P according to the question. Setting the encoder as f, and the trainable parameters



Knowledge Integration Instruct Prompt I : (Learning Knowledge)
Instruction: (None)
Input: Walter Matthau really looks like a
Output: skid row bum! Matthau was a respected ... He made his motion picture debut as a whip-wielding bad guy in The Kentuckian (1955) ... , he made Ride a Crooked Trail with Audie Murphy, and Onionhead (both 1958) starring Andy Griffith; the latter was a flop. Matthau had a featured role opposite Griffith ...

Knowledge Integration Instruct Prompt II : (Answering Question)
Instruction: Please answer the question according to the text.
Input: Text: Walter Matthau really looks like askid row bum! Matthau was a respected ... He made his motion picture debut as a whip-wielding bad guy in The Kentuckian (1955) ... Ride a Crooked Trail with Audie Murphy, and Onionhead (both 1958) starring Andy Griffith; the latter was a flop. Matthau ... Question: What was Walter Matthau's first movie?
Output: The Kentuckian

Controllabe Generation Instruct Prompt III: TriviaQA,CMRC :
Instruction: Please extract the answer keyword.
Input: Question: What was Walter Matthau's first movie? Answer: The Kentuckian movie in 1950s.
Output: Processed answer: The Kentuckian

MSMARCO :
Instruction: Please refine the response to make it more closely aligned with the question. Input: Question: What was Walter Matthau's first movie?
Answer: It is The Kentuckian. Output: Processed answer: Walter Matthau's first movie is The Kentuckian.

Controllabe Generation Instruct Prompt IV:
Instruction: You will serve as an assistant to help me determine which answer is exactly correct, brief and coherent. Please use the word exactly from the choices to answer.
Input: Question: What was Walter Matthau's first movie?
The answers to choose from are [The Kentuckian, The Kentuckian movie in 1950s.]
Output: Processed answer: The Kentuckian

Fig. 2: The overall framework of GenKI. The left part presents several examples of our prompts used in the knowledge integration and controllable generation module. The right part outlines the process of tuning LLMs to learn knowledge and answer questions in specific task situations.

as Φ, DPR computes question-sentence similarity for a given question q and a set of sentences P using the formula below:

$$S_{q,p;\Phi} = f_q(q, \Phi_q)^T f_p(p; \Phi_p),\ \ p \in P, \tag{2}$$

where $\Phi = [\Phi_q, \Phi_p]$ denotes the retriever question and passage encoder parameters. We choose the final retrieved Top-k passage as below:

$$P_R = R(P, q) = \{p_1, ..., p_k\} = Top_k(min(S_{q,p;\Phi}, p \in P)). \tag{3}$$

The retrieved result in P_R will be used as input and tuning material for knowledge integration, which can provide LLMs with more accurate knowledge.

4.2 Knowledge Integration

From the previous part, we acquired high-quality knowledge utilizing the retriever. To integrate this high-quality knowledge into the parameters of our model, tuning is an effective way. However, the original instruction tuning is inadequate for the OpenQA task, which is attributed to its limited scope of only aiding LLMs in comprehending the objectives of different tasks. Hence the knowledge integration $A_K = K(q, P)$ aims to enable the model to receive the necessary knowledge for generating the target answer. The detailed demonstration is shown in Figure 2 part A.

Inspired by how pretraining enables LLMs to grasp knowledge [7,15,47], we also introduce the following tuning optimization to enable the model to learn more domain knowledge precisely. However, as one of the innovations of this paper, our Knowledge Integration module is neither learning all knowledge extensively like pre-training nor providing knowledge only in the prompt like contextual learning. The fine-tuning paradigm designed in this paper integrates retrieved refined knowledge into LLM through fine-tuning, ensuring the accuracy and domain specificity of the knowledge.

Specifically, tuning an LLM on domain knowledge P is defined as follows:

$$\mathcal{L}_r = max_{\Phi} \sum_{p \in P} \sum_{t=1}^{|p|} log(P_{\Phi}(p_t | p_{<t})). \tag{4}$$

The final optimization loss is as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_r + \lambda_2 \mathcal{L}_f = max_{\Phi_L}(\lambda_1 \sum_{p \in P} \sum_{t=1}^{|p|} log(P_{\Phi+\Phi_L}(p_t | p_{<t})) + \lambda_2 \mathcal{L}_f, \tag{5}$$

where $\mathcal{L}_f$ is the loss of original instruction tuning, Φ_L is the LoRA parameters and we only update LoRA parameters during the training process. The prompts we used are shown in Figure 2. In the experiment, we set $\lambda_1 > \lambda_2 > 0$ to make the model focus on knowledge without losing the ability to answer questions.

However, due to the occasional presence of inaccuracies in the initial retrieval results K_R leading mistake of $A_{K_R} = K(q, P_R)$, we adopt this module with the input of either full passage P or retrieved knowledge P_R using $A_K = K(q, P)$ or $A_{K_R} = K(q, P_R)$. These output options are presented to the controllable generation step, allowing for a selective choice process.

Incorporating knowledge during finetuning using this method leverages the advantages of the retriever, enabling the model to learn more precise and specialized knowledge. By storing domain knowledge in LLMs' parameters, our work leads the model to learn knowledge more stably, resulting in superior performance. This method is used in conjunction with prompts using retrieved results to further enhance the model's capability.

4.3 Controllable Generation

From the knowledge integration model, we gain a draft of the answer. While the draft of this answer already contains all the knowledge needed to address the question, it still does not meet our goal of formatted generation. In order to align the model's output with the dataset's answers and our expectations, we need to introduce another framework for post-processing the answers from the last module, formally defined as $C(A_K) = A$.

The approach employed for achieving controllable generation utilizes the identical loss function as instruction tuning. However, due to the fact that the preceding step has yielded answers imbued with high-quality knowledge, aligning them into the format we require becomes a more manageable task than

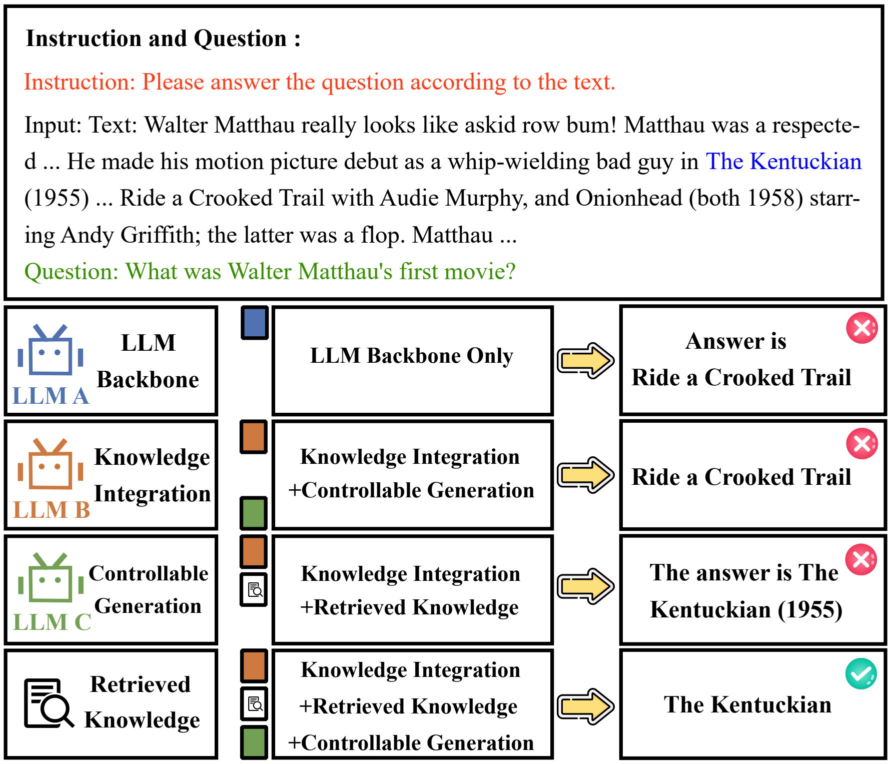

Instruction and Question :
Instruction: Please answer the question according to the text.
Input: Text: Walter Matthau really looks like askid row bum! Matthau was a respected ... He made his motion picture debut as a whip-wielding bad guy in The Kentuckian (1955) ... Ride a Crooked Trail with Audie Murphy, and Onionhead (both 1958) starring Andy Griffith; the latter was a flop. Matthau ...
Question: What was Walter Matthau's first movie?
LLM A
LLM Backbone
LLM B
Knowledge Integration
LLM C
Controllable Generation
Retrieved Knowledge
LLM Backbone Only
Knowledge Integration +Controllable Generation
Knowledge Integration +Retrieved Knowledge
Knowledge Integration +Retrieved Knowledge +Controllable Generation
Answer is Ride a Crooked Trail
Ride a Crooked Trail
The answer is The Kentuckian (1955)
The Kentuckian

Fig. 3: The real cases of outputs after Knowledge Integration and Controllable Generation structure.

instruction tuning for the model to undertake. After fine-tuning the model with a certain amount of training data, we obtain the post-processing model Pp. The instruction we used for fine-tuning is demonstrated in part B in Figure 2.

Additionally, derived from the previous step, we have acquired two distinct answers denoted as $Pp(A_K)$ and $Pp(A_{K_R})$. The imminent task involves the meticulous selection of one among these to serve as the ultimate output. In order to make our answer reliable and controllable, we combine a reward model to effectively select answers that match the output style of the dataset and an external judging model when these two answers are severely different. The pairwise loss of the reward model chooses one best answer among the two as below:

$$\mathcal{L}rm = -log(\sigma(R_m(A_+) - R_m(A_-)), \tag{6}$$

where $Rm(A)$ is the average score of the embedding of the final token of A. To encourage answer format alignment and coherence enhancement when answers are long and similar, we then design the judgment criteria function as below:

$$S = Exp(|C_s(QA_1) - C_s(QA_2)|) - (|R_m(A_1) - R_m(A_2)|)/\overline{R_m(A)} * \overline{len(A)}, \tag{7}$$

where Q denotes questions, A_1 and A_2 respectively represents $Pp(A_K)$ and $Pp(A_{K_R})$. $len(A)$ is the length of answers and $\overline{len(A)}$ is the mean of A. Operations like Exp aim to make C_S score and R_m score at the same scale, while $len(A)$ is designed to make model focus more on fluency when the answer is longer. We adopt normalized inverse sentence frequency (NISF)[25] to calculate answer consistency C_S, which is formally calculated as below:

$$NISF(Y_j) = \frac{ISF(Y_j)}{\Sigma_{k=1}^{M} ISF(Y_k)}, \tag{8}$$

$$C_s(QA) = NISF(A) log P_\theta(A|Q[M]) + NISF(Q) log P_\theta(Q|[M]A), \tag{9}$$

where Q, $\{Y_1, ..., Y_j\}$ denotes the question and the sentences of the question-answer pair. $[M]$ demotes masking when calculating probability using LM and A denotes the answer. $ISF(Y)$ the inverse sentence frequency, which is related to the maximum of maximum inverse word frequency (IWF) below:

$$IWF(\omega) = \frac{log(1 + |C|)}{f_\omega}, \tag{10}$$

$$ISF(Y_j) = max_{\omega \in Y_j} IWF(\omega), \tag{11}$$

where $\{\omega_{11}, ..., \omega_{1i}, ..., \omega_{ji}\}$ denotes the words of the sentences. $|C|$ indicates the total number of sentences, f_ω denotes the frequency of ω. After computing the score, we can employ the reward model to ensure answer coherence and format alignment or prioritize the correctness and reliability of answers derived through Ext, an external selection using chatGPT. Despite chatGPT's performance not exceeding our results from the previous stage, it plays a role in augmenting overall capability. Additional details are presented in Section 6.3. The overall algorithm is delineated in Algorithm 1.

By combining instruction-tuned LLM and a controllable answer selection model, we will gain an answer with both fluency brought by the reward model and answer-format accuracy brought by instruction-tuned LLM. Finally, to provide a clearer demonstration of how different components of GenKI contribute to its performance, we present real cases of GenKI's outputs at different stages in Figure 3. This visualization allows us to observe GenKI's transformation and refinement of outputs as it progresses through each stage of the process.

5 Experimental Settings

We present the datasets, baselines, and evaluation metrics for comparing with baselines and evaluating the effectiveness of our model below.

Algorithm 1 Overall pseudo code of GenKI

```
Input: Knowledge base $K$, Retriever $R$, LLM $L$, question set $Q$
Output: Answer $A$
Train $L_1$ and $L_2$ with $K$ and $K_R = R(K, Q)$ respectively    ▷ Prompt I, II is used here
Gain $P_3$ using $L_2$ answering training queries
Train $L_3$ with $P_3$    ▷ Prompt III is used here
$A_K = L_1(Q), A_{K_R} = L_2(Q)$
$A_1 = P_p(A_K) = L_3(A_K), A_2 = P_p(A_{K_R}) = L_3(A_{K_R})$
$S = Exp(|C_s(A_1) - C_s(A_2)|) - (|R_m(A_1) - R_m(A_2)|)/\overline{R_m(A)} * \overline{len(A)}$
if S<0 then
    $A = max_A(Rm(Pp(A_K)), Rm(Pp(A_{K_R})))$
else
    $A = Ext(Pp(A_K), Pp(A_{K_R}))$    ▷ Prompt IV is used here
end if
```

5.1 Datasets

To demonstrate the performance of our proposed model on different answer formats of OpenQA, we conduct experiments on three publicly aware datasets, **TriviaQA** [22], **MSMARCO** [4], and **CMRC-2018** [11], which not only represent different answering format (free answer or span answer from the passage) but also include various answering lengths (one or two words or a sentence). In terms of data pre-processing, we filter out the data without correct answers on the MSMARCO dataset and randomly sample the same amount of data as TriviaQA to comparably evaluate performances. The detailed description and statistics of these datasets[3] can be referred to in Table 1.

Specifically, **TriviaQA** is a reading comprehension dataset containing over 650K training question-answer-evidence triples. The answers are mostly composed of entities in Wikipedia, containing one or two words. **MSMARCO** dataset contains 8,841,823 passages from documents retrieved by the Bing search engine, which provides the necessary information for curating the natural language answers. Different from TriviaQA, these answers belong to long and fluent sentences. **CMRC-2018** is a dataset for Chinese Machine Reading Comprehension. It is a span-extraction reading comprehension dataset, which demands answers completely extracted from the original text. Following [16], we processed reading comprehension datasets (CMRC and MSMARCO) by shuffling their reference, aligning the formal definition of the OpenQA, where knowledge must be searched from the knowledge base, not given.

5.2 Baselines

To comprehensively compare and demonstrate the effectiveness of GenKI, we have chosen three different baselines. Task-specific baselines are the most competitive baselines on each dataset differently to make a fair comparison because

[3] The processed dataset will be open once accepted.

different from GenKI, as these task-specific methods can not achieve consistent performance across all datasets. The latest baselines are essentially based on LLMs, as methods relying on LLMs consistently outperform traditional approaches. LLM backbones are the raw model used in our structure, demonstrating our enhancement to the basic model. The detailed baseline settings will be illustrated as follows:

Table 1: The detailed description and statistics of datasets.

Dataset	**Format**	**Length**	**QA pairs**
TriviaQA	Free	Entity	7993
MSMARCO	Free	Sentence	8000
CMRC-2018	Span	Entity/Sentence	3219

TriviaQA Specific Baselines The baselines we choose to compare with our model in TriviaQA is the top three methods[4] that share a similar scale with ours. This is because comparing our model with methods having a significantly larger scale of parameters would be unfair, as they require more time and computational resources for training than we do. These three methods include:

- **ChatGPT** leverages a transformer-based neural network to understand and generate human language. We choose the gpt-3.5-turbo-0301 version as a baseline for its stable performance.
- **Codex+REPLUG** [43] employs retrieval systems to fetch relevant documents as prompt.
- **GLaM-62B** [14] uses a sparsely activated mixture-of-experts architecture to parallelize model.

MSMARCO Specific Baselines We select the top-3 models on MSMARCO question answering Natural Language Generation Task[5] as baselines. These three methods include:

- **PALM** [6] alleviates the mismatch between pre-training and fine-tuning where generation is more than reconstructing the original text.
- **Masque NLGEN Style** [37] propose a multi-style summarization model regarding question answering as summarizing on question and reference.
- **REAG** is an anonymous submit and gains third place.

[4] https://paperswithcode.com/sota/question-answering-on-triviaqa

[5] https://microsoft.github.io/msmarco/

CMRC Specific Baselines We also select the top-performing 3 models on CMRC-2018[6] as baselines, which include:

- **ERNIE-Gram** [51], an explicit n-gram masking method instead of token masking training.
- **MacBERT** [10] utilizes MLM as a correction task to train and mitigate the discrepancy with Bidirectional Encoder Representations from Transformers.
- **ERNIE2.0** [45] captures lexical, syntactic and semantic aspects of information in the training data.

Latest Baselines Despite these top-performing methods, we also adopt four latest LLM-based structures on OpenQA as baselines, following [17] we classify them into retriever-augmented baselines (the first two) and generator-augmented baselines (the last two):

- **LLM-KB** [40] uses LLM itself to augment the reference to enhance the zero-shot learning effect.
- **LLaMA2-index** [7], an end-to-end alignment and rerank structure to merge align Retriever with LLM, which benefit from context augmentation.
- **RFiD** [48] finetunes the decoder to generate answers by to differentiate between causal relationships and spurious features.
- **Self-RAG** [3] fine-tuned an LLM generates, reflects, and critiques retrieved passages and their generations using special tokens.

LLM backbone In order to further demonstrate the improvement of our framework GenKI, we also adopt the LLM backbones as baselines. We use LLaMA-65B and GLM-6B as pre-trained LLM backbones, which allows us to explore different-scale models and also test the effectiveness of our framework from both perspectives of English and Chinese.

5.3 Evaluation Metrics

Different datasets employ various evaluation methods for assessing answers. For datasets like TriviaQA and CMRC-2018, which emphasize consistency with the answers, we use the Exact Match (EM), F1 score, and Recall as evaluation metrics. Specifically, in this experiment, the dataset consists of multiple answers, each of which is considered a correct answer. Therefore, the Recall and F1 metrics in this experiment differ from those of multiple-choice questions and are instead based on text-level Recall and F1. For example, if the correct answer is "large language model," and the generated answer is "language model," the Recall value would be 2/3. The final score for a question is determined by taking the highest value among all answers. To illustrate that our framework GenKI is capable of enabling models to generate fluent, extended content, we utilize BLEU and

[6] https://paperswithcode.com/sota/chinese-reading-comprehension-on-cmrc-2018
[7] https://www.llamaindex.ai

ROUGE metrics to evaluate the quality of answers in the MSMARCO dataset. Moreover, to gauge the fluency of model outputs, we employ the Coherence values derived from CTRLEval[26]. We categorize these metrics into two categories, K and C, where metrics in the K class assess the model's knowledge integration capability, and metrics in the C class measure the model's controllable generation ability. Among them, EM and f1 in TriviaQA and CMRC, BLEU and ROUGE in MSMARCO are in category K, while EM in CMRC, EM in TriviaQA and Coherence value in MSMARCO are in category C.

5.4 Implementation Details

The instruction-tuning and model inference are conducted on 2 Tesla A100 80G GPUs for GLM-6B and 8 Tesla A800 80G GPUs for LLaMA-65B. Across all methods, we finetune the model with LoRA[19] with Lora-rank 32 in GLM and 8 in LLaMA. The model parameters are optimized using the Adam[27] with a default learning rate of 1e-4. We trained LLMs using the LoRA method to obtain three sets of fine-tuned parameters, each proposed for implementing full knowledge integration, retrieved knowledge integration and controllable generation. The total parameters of our structure were about 1(frozen LLM backbone) + 3*6%(LoRA tunable parameters), altogether 7.08B in GLM and 76.7B in LLaMA. Furthermore, to ensure fair comparisons, we tuned the parameters of all baseline models to their best performance. All the generator-augmented baselines (finetuned-LLM baselines) enjoy the same type of instruction tuning.

6 Results and Analysis

In this section, we first present a performance comparison between our method and the baseline across three datasets. Then, through a detailed analysis, we demonstrate how the two main modules in GenKI, i.e., Knowledge Integration and Controllable Generation, work for OpenQA on LLMs.

6.1 Overall Performance

We present the overall performance of GenKI and all baselines in Tables 2, 3 and 4 on TriviaQA, MSMARCO, and CMRC, respectively.

In **TriviaQA**, the LLaMA model, optimized with our framework, successfully achieved a 4.8% gain on the EM metric and a 5.6% gain on the F1 metric compared to the backbone itself. The GLM model, optimized with our framework, successfully gained 891.7% and 277.8% on the EM and F1 metrics, respectively, compared to the backbone itself. Specifically, the LLaMA model is compared against the current retrieval-augmented (REPLUG, LLM-KB, LLaMA2-index), generator-augmented (RFiD, Self-RAG) and few-shot learning (ChatGPT, GLaM-62B) approaches in OpenQA, outperforming them by 4.7% and 3.6% on the EM and F1 metrics, respectively. Moreover, it achieved comparable performance to ChatGPT, which is currently regarded as one of the best LLMs.

Table 2: OpenQA performance of EM and F1-value on TriviaQA dataset.

Baseline Type		**Method**	**EM%**	**F1%**
Task Specific Baselines		GLaM-62B	75.8	-
		Codex+REPLUG	77.3	-
		Gpt-3.5-turbo-0301	77.9	83.9
Latest Baselines	RA	LLM-KB	74.8	80.1
		LLaMA2-index	67.4[8]	-
	GA	RFiD	72.7	-
		Self-RAG	69.3	-
LLM Backbones		GLM-6B$_{base}$	7.3	20.8
		LLaMA-65B$_{base}$	77.8	82.3
Our Work		GLM-6B$_{GenKI}$	72.4	78.6
		LLaMA-65B$_{GenKI}$	**81.6**	**86.9**

RA: retriever-augmented baselines, *GA*: generator-augmented baselines (finetuned-LLM baselines)

Table 3: OpenQA performance of BLEU and ROUGE on MSMARCO dataset.

Baseline Type		**Method**	**BLEU-1**	**ROUGE-L**
Task Specific Baselines		Masque NLGEN	0.501	0.496
		PALM	0.499	0.498
		REAG	0.497	**0.498**
Latest Baselines	RA	LLM-KB	0.262	0.241
		LLaMA2-index	0.588	0.447
	GA	RFiD	0.561	0.454
		Self-RAG	0.567	0.351
LLM Backbones		LLaMA-65B$_{base}$	0.473	0.278
		GLM-6B$_{base}$	0.475	0.248
Our Work		LLaMA-65B$_{GenKI}$	0.521	0.317
		GLM-6B$_{GenKI}$	**0.598**	0.456

RA: retriever-augmented baselines, *GA*: generator-augmented baselines (finetuned-LLM baselines)

In the **MSMARCO** dataset, the GLM model, optimized with our framework, outperforms the backbone itself by 25.9% and 83.8% on the BLEU-1 and ROUGE-L metrics. However, LLaMA falls short compared to GLM in terms of performance. This could possibly be attributed to its lack of a supervised training process, which leads to difficulties in generating lengthy and coherent text. Even so, LLaMA still shows improvement by 10.1% and 14.0% on the BLEU-1 and ROUGE-L metrics. Our GenKI framework outperforms the best model by a margin of 1.7% on the BLEU-1 metric. Nevertheless, in our framework, the

ROUGE-L metric lagged behind the performance of the current top-performing approach. This discrepancy arises due to the fact that the references in the MS-MARCO dataset comprise lengthier passages that closely resemble the source text, contrasting with knowledge integration capacity of our framework.

Within the **CMRC** dataset, the GLM model outperforms the backbone itself by a significant margin, exceeding the baseline performance by over 40 times. Meanwhile, the GLM model shows improvements over the existing state-of-the-art approach by 5.4% and 0.9% on the EM and F1 metrics, respectively. This accomplishment is particularly remarkable given that CMRC-2018 operates as a sequence labeling dataset. Generative models inherently encounter challenges in this context, as they must generate results that align precisely with the source text—a requirement that inherently presents a natural disadvantage. We refrained from employing the LLaMA and RFiD models on this dataset due to their inadequate performance in handling the Chinese language. To further analyze and evaluate the effectiveness of our model, we conducted the following analyses on two special cases:

Can GenKI work when the entire knowledge base is inaccessible? In practical academic and industry contexts, models frequently depend on search engines to gather requisite knowledge, facing limitations in leveraging the entire knowledge base for fine-tuning. In such instances, our model $(X + K_R + P_p)$ consistently surpasses baseline performances, demonstrating superiority by 3.3% in TriviaQA, 0.8% in MSMARCO, and 4.9% in CMRC. The comparative results are illustrated in Figure 5.

Can GenKI adapt to different formats of answers? To demonstrate the adaptability of our model to another format domain, we partition the CMRC dataset, encompassing both long and short-answer formats by their average length. Specifically, we fine-tune our model only on the short-answer segment during step P_p, resulting in a noteworthy result of 71.70 on the EM metric. This achievement surpasses all recent baselines, trailing behind only one of the task-specific baselines. Notably, given that these task-specific baselines are exclusively trained intra-domain, our model exhibits exceptional performance, underscoring its remarkable out-of-domain capability. The comparison is shown in Figure 5.

To sum up, analysis of these three datasets demonstrates that GenKI has performed remarkable ability in learning retrieved knowledge and multi-format controllable generation compared with task-specific, latest baselines both on retriever and generator augmentation and LLM backbones.

6.2 Analysis on Knowledge Integration module

To what extent can the acquired knowledge impact the proficiency of the model? In this section, we set aside all output format requirements and focus solely on studying the model's grasp and understanding of knowledge. The TriviaQA and CMRC datasets, which are more relevant in terms of knowledge,

Table 4: OpenQA performance of EM and F1-value on CMRC dataset.

Baseline Type		**Method**	**EM%**	**F1%**
Task Specific Baselines		ERNIE2.0	69.1	88.6
		MacBERT	70.7	88.9
		ERNIE-Gram	74.3	90.5
Latest Baselines	RA	LLM-KB	24.3	48.8
	GA	RFiD	43.7	65.1
LLM Backbones		GLM-6B$_{base}$	1.8	47.7
Our Work		GLM-6B$_{GenKI}$	**78.3**	**91.1**

RA: retriever-augmented baselines, *GA*: generator-augmented baselines (finetuned-LLM baselines)

are chosen to facilitate the investigation of this ability better. Additionally, we adopt the weakest format requirement, recall value (where the model mentions words in the answer), for our study. As the ablation results are shown in Table 5, we observe an improvement of recall by 19.4% and 6.2% in TriviaQA for the GLM and LLaMA models, respectively, after applying K_R. To ascertain the link between this enhancement and the quality of the retrieval results, we conduct comparative experiments under varying retrieval quality levels.

We measure the quality of retrieval results by utilizing the frequency of occurrence of the ground truth within the retrieval results. For instance, if the ground truth is "Large Language Model" and our retrieval result is "Large language models have gained widespread language applications." then the quality of the retrieval result would be calculated as $3/8 = 0.375$, without calculating the latter occurrence of "language". Subsequently, the effectiveness of the model's integration of knowledge is evaluated using the Recall metric. Additionally, we constrain the maximum output length of the model to prevent the model from achieving a high recall by producing excessively redundant results. We then select a subset of examples from TriviaQA, consisting of both higher-quality and lower-quality retrievals, to gain more data for studying the impact of retrieval quality on the model's generated recall values.

The relationship is demonstrated in Figure 4. From this figure, we can infer two pieces of information:

(1) The relationship between retrieval quality and model recall initially exhibited a linear trend across these examples ($R^2 > 0.99$), but eventually reached a bottleneck state, transitioning into another linear relationship ($R^2 > 0.985$). This suggests that the model has been extensively integrated with the necessary knowledge. The ultimate bottleneck of the model is correlated with the scale of the model. For instance, the bottleneck for LLaMA-65B occurred at around 90% recall, while the bottleneck for GLM-6B was observed at around 80% recall.

(2) By comparing the recall of the LLM backbone with the recall of the LLM integrated with full knowledge($X + K$ on table 5, 6), or whether the model

Table 5: Ablation Experimental Results on TriviaQA.

TriviaQA							
Metric category	K&C	K	K		K&C	K	K
X=GLM-6B	EM%	F1%	Recall%	X=LLaMa-65B	EM%	F1%	Recall%
X	7.3	20.8	58.3	X	77.8	82.3	82.5
$X+K$	47.4	61.7	66.3	$X+K$	72.7	79.0	83.7
$X+K+Pp$	57.1	64.7	64.9	$X+K+Pp$	75.9	80.9	82.8
$X+K_R$	54.0	66.2	69.6	$X+K_R$	72.7	80.1	87.6
$X+K_R+Pp$	58.0	66.6	67.2	$X+K_R+Pp$	80.5	85.4	86.9
Rm	58.6	68.2	69.0	Rm	80.3	85.4	85.6
Ext	72.4	78.6	79.8	Ext	81.5	86.5	88.7
$GenKI$	72.4	78.6	79.9	$GenKI$	81.6	86.7	87.2

* X: LLM backbone, K:output after full knowledge integration model, K_R: output after retrieved knowledge integration model

** Pp: output after controllable generation, Rm: output after reward model, Ext: external model selection

reaches a bottleneck, we can infer whether the pretraining of the model includes the required knowledge of target dataset.

First, it can be inferred that the GLM's pre-training hardly contains Wikipedia knowledge (58.3 compared to 66.3), but found a bottleneck in the CMRC from the beginning. This observation aligns with our expectations, given that GLM focuses more on Chinese language corpora. Additionally, it can be inferred that LLaMA's pretraining contains knowledge from Wikidata(82.5 similar to 83.7), which is also supported by findings in LLaMA's technical report [47].

Table 6: Ablation Experimental Results on MSMARCO and CMRC.

MSMARCO					CMRC-2018			
Metric category	K&C	K&C	K&C	C		K&C	K	K
X=GLM-6B	BLEU-1	BLEU-2	ROUGE-L	Coherence	X=GLM-6B	EM%	F1%	Recall%
X	0.475	0.387	0.248	-4.164	X	1.8	47.7	92.1
$X+K$	0.588	0.523	0.428	-4.195	$X+K$	33.5	71.8	93.1
$X+K+Pp$	0.589	0.526	0.455	-4.185	$X+K+Pp$	77.8	91.0	92.3
$X+K_R$	0.594	0.526	0.429	-4.196	$X+K_R$	37.1	73.3	93.8
$X+K_R+Pp$	0.593	0.529	0.452	-4.198	$X+K_R+Pp$	78.0	91.1	92.2
Rm	0.597	0.531	0.457	-4.184	Rm	78.1	91.3	92.8
Ext	0.598	0.534	0.457	-4.195	Ext	75.2	89.9	91.4
$GenKI$	0.598	0.534	0.456	-4.174	$GenKI$	78.3	91.1	92.3

* Coherence in CTRLEval

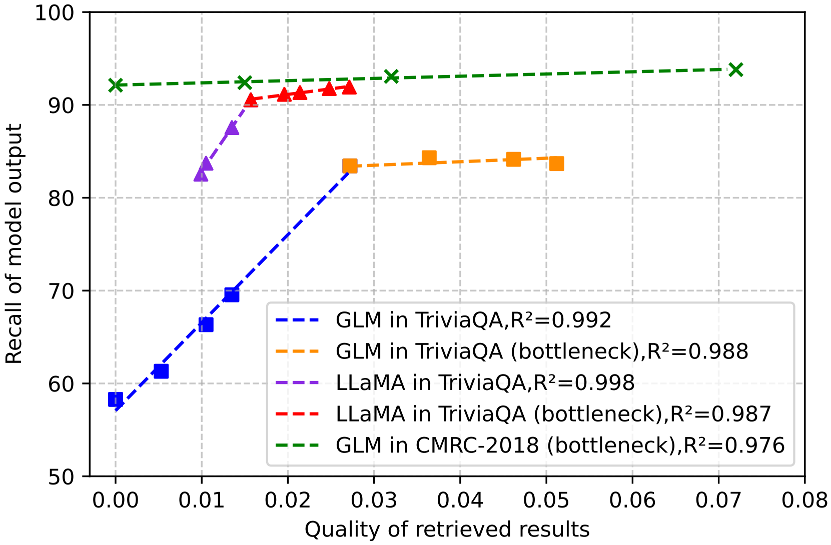


Fig. 4: The linear fitting relationship between quality of retrieved result and knowledge integration effect

However, merely enhancing the model's knowledge understanding through the Knowledge Integration module is insufficient. Overemphasizing the model's knowledge understanding may weaken its controllable generation ability. Specifically, the performance of both $X + K$ and $X + K_R$ is better than X when using GLM-6B but worse when using LLaMa-65B. This is also why our work designs a novel three-stage paradigm of retrieval, knowledge integration, and controllable generation, avoiding the distributional differences of LLM in the two-stage RAG. Next, this paper will analyze the controllable generation module.

6.3 Analysis on Controllable Generation Module

Can GenKI tackle distribution between knowledge integration or controlled generation? The success of our model lies in effectively balancing both knowledge integration and controlled generation abilities. To substantiate this claim, we categorize the metrics into two components: knowledge integration metric K and controlled generation metric C. As shown in table 5 and table 6 our model succeeds in obtaining all $K\&C$ and C metrics and most K metrics.

We are also concerned about how different structures influence our answers. After undergoing Step Pp in Table 5, the recall of all answers decreases (by up to 2%), while the EM of all answers increases(by up to 2 times). This indicates that the model is more inclined to have learned distribution of instructions which is different from knowledge. This observation underscores the importance of splitting the answer generation process into two steps: knowledge integration and controllable generation. The Ext step plays the same effect on the recall of answers. It performs better when the backbone LLM performs worse. The improvement reaches 18.8% recall in TriviaQA using GLM-6B.

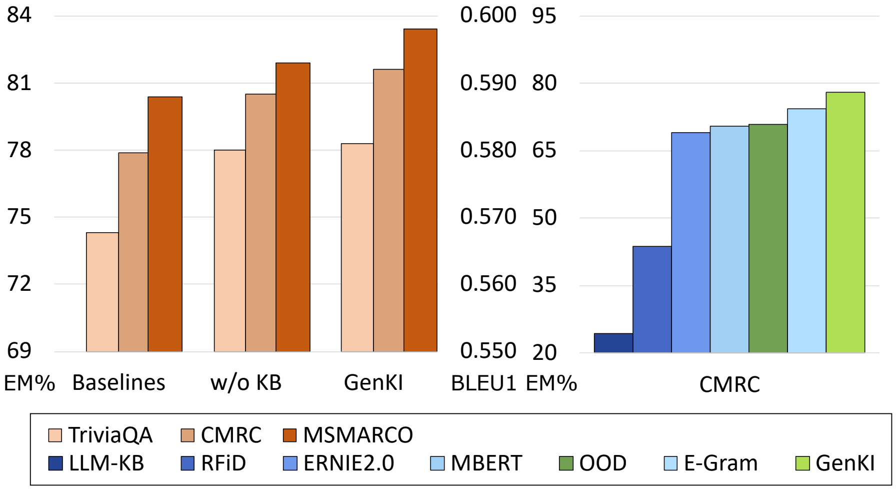


Fig. 5: Comparison of (Left) GenKI, GenKI without knowledge base and best baseline, (Right) GenKI, GenKI in Out-Of-Domain scenario and baselines.

The Reward Model step, on the other hand, demonstrates improvements in coherence(by 1.8%) within the MSMARCO dataset, as shown in Table 6. Our Reward Model module achieved better results in all datasets, except it slightly loses efficacy on LLaMA in TriviaQA, which aims to enhance answer coherence and consistency to more closely align with human preferences.

The introduction of the external model selection module has achieved significant success on knowledge question-answering datasets such as CMRC and TriviaQA. Particularly, for models with weaker knowledge mastery like GLM-6B, there is a substantial improvement (24.8% in EM-score). This indicates that the introduction of this module can effectively ensure the accuracy of model knowledge. Additionally, our data also confirms its compatibility with our framework.

7 Conclusion

In this paper, we introduced GenKI, a novel OpenQA framework that combines Knowledge Retrieval, Knowledge Integration, and Controllable Generation. By splitting the generation process of LLMs into two distinct phases of knowledge integration and controllable generation, we addressed the challenges of knowledge integration deficiency and answer format misalignment. GenKI has demonstrated enhanced performance in OpenQA tasks, surpassing both traditional methods and other LLM-based approaches. This research not only explores the relationship between provided search results and knowledge generation within LLMs but also propels LLM-based OpenQA towards the desired format.

References

1. Abdallah, A., Jatowt, A.: Generator-retriever-generator: A novel approach to open-domain question answering. arXiv preprint arXiv:2307.11278 (2023)
2. Aizawa, A.: An information-theoretic perspective of tf–idf measures. Information Processing & Management **39**(1), 45–65 (2003)
3. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-rag: Learning to retrieve, generate, and critique through self-reflection. arXiv preprint arXiv:2310.11511 (2023)
4. Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T.: Ms marco: A human generated machine reading comprehension dataset (2018)
5. Bevilacqua, M., Ottaviano, G., Lewis, P.S.H., Yih, S., Riedel, S., Petroni, F.: Autoregressive search engines: Generating substrings as document identifiers. In: NeurIPS (2022), http://papers.nips.cc/paper_files/paper/2022/hash/cd88d62a2063fdaf7ce6f9068fb15dcd-Abstract-Conference.html
6. Bi, B., Li, C., Wu, C., Yan, M., Wang, W., Huang, S., Huang, F., Si, L.: Palm: Pre-training an autoencodingautoregressive language model for context-conditioned generation (2020)
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
8. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways. CoRR **abs/2204.02311** (2022). https://doi.org/10.48550/arXiv.2204.02311, https://doi.org/10.48550/arXiv.2204.02311
9. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
10. Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G.: Revisiting pre-trained models for Chinese natural language processing. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 657–668. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.58, https://aclanthology.org/2020.findings-emnlp.58
11. Cui, Y., Liu, T., Che, W., Xiao, L., Chen, Z., Ma, W., Wang, S., Hu, G.: A span-extraction dataset for Chinese machine reading comprehension. 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). pp. 5886–5891. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1600, https://www.aclweb.org/anthology/D19-1600

12. Das, R., Dhuliawala, S., Zaheer, M., McCallum, A.: Multi-step retriever-reader interaction for scalable open-domain question answering. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=HkfPSh05K7
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
14. Du, N., Huang, Y., Dai, A.M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A.W., Firat, O., et al.: Glam: Efficient scaling of language models with mixture-of-experts. In: International Conference on Machine Learning. pp. 5547–5569. PMLR (2022)
15. Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., Tang, J.: Glm: General language model pretraining with autoregressive blank infilling. arXiv preprint arXiv:2103.10360 (2021)
16. Dua, D., Strubell, E., Singh, S., Verga, P.: To adapt or to annotate: Challenges and interventions for domain adaptation in open-domain question answering. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 14429–14446. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.807, https://aclanthology.org/2023.acl-long.807
17. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)
18. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D.d.L., Hendricks, L.A., Welbl, J., Clark, A., et al.: Training compute-optimal large language models. arXiv preprint arXiv:2203.15556 (2022)
19. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
20. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. CoRR **abs/2007.01282** (2020), https://arxiv.org/abs/2007.01282
21. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: Spanbert: Improving pre-training by representing and predicting spans. Transactions of the association for computational linguistics **8**, 64–77 (2020)
22. Joshi, M., Choi, E., Weld, D., Zettlemoyer, L.: 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). pp. 1601–1611. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1147, https://aclanthology.org/P17-1147
23. Kandpal, N., Deng, H., Roberts, A., Wallace, E., Raffel, C.: Large language models struggle to learn long-tail knowledge. In: International Conference on Machine Learning. pp. 15696–15707. PMLR (2023)
24. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550, https://aclanthology.org/2020.emnlp-main.550

25. Ke, P., Zhou, H., Lin, Y., Li, P., Zhou, J., Zhu, X., Huang, M.: CTRLEval: An unsupervised reference-free metric for evaluating controlled text generation. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2306–2319. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.164, https://aclanthology.org/2022.acl-long.164
26. Ke, P., Zhou, H., Lin, Y., Li, P., Zhou, J., Zhu, X., Huang, M.: Ctrleval: An unsupervised reference-free metric for evaluating controlled text generation (2022)
27. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
28. Lewis, P.S.H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
29. Li, J., Zhang, Z., Zhao, H.: Self-prompting large language models for zero-shot open-domain qa (2023)
30. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
31. Ma, X., Gong, Y., He, P., Zhao, H., Duan, N.: Query rewriting for retrieval-augmented large language models. arXiv preprint arXiv:2305.14283 (2023)
32. Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., Hajishirzi, H.: When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In: Rogers, A., Boyd-Graber, J.L., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023. pp. 9802–9822. Association for Computational Linguistics (2023). https://doi.org/10.18653/v1/2023.acl-long.546, https://doi.org/10.18653/v1/2023.acl-long.546
33. Mao, Y., He, P., Liu, X., Shen, Y., Gao, J., Han, J., Chen, W.: Generation-augmented retrieval for open-domain question answering. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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). pp. 4089–4100. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.316, https://aclanthology.org/2021.acl-long.316
34. Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., Roth, D.: Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys (2021)
35. Min, S., Chen, D., Zettlemoyer, L., Hajishirzi, H.: Knowledge guided text retrieval and reading for open domain question answering. CoRR **abs/1911.03868** (2019), http://arxiv.org/abs/1911.03868
36. Nguyen, H.H., Liu, Y., Zhang, C., Zhang, T., Yu, P.S.: Cof-cot: Enhancing large language models with coarse-to-fine chain-of-thought prompting for multi-domain nlu tasks. arXiv preprint arXiv:2310.14623 (2023)
37. Nishida, K., Saito, I., Nishida, K., Shinoda, K., Otsuka, A., Asano, H., Tomita, J.: Multi-style generative reading comprehension (2019)

38. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)
39. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.: Language models as knowledge bases? 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). pp. 2463–2473. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1250, https://aclanthology.org/D19-1250
40. Ren, R., Wang, Y., Qu, Y., Zhao, W.X., Liu, J., Tian, H., Wu, H., Wen, J.R., Wang, H.: Investigating the factual knowledge boundary of large language models with retrieval augmentation. arXiv preprint arXiv:2307.11019 (2023)
41. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (apr 2009). https://doi.org/10.1561/1500000019, https://doi.org/10.1561/1500000019
42. Sachan, D.S., Patwary, M., Shoeybi, M., Kant, N., Ping, W., Hamilton, W.L., Catanzaro, B.: End-to-end training of neural retrievers for open-domain question answering. arXiv preprint arXiv:2101.00408 (2021)
43. Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., Yih, W.t.: Replug: Retrieval-augmented black-box language models. arXiv preprint arXiv:2301.12652 (2023)
44. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021. pp. 3784–3803. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.320, https://doi.org/10.18653/v1/2021.findings-emnlp.320
45. Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., Wang, H.: Ernie 2.0: A continual pre-training framework for language understanding. Proceedings of the AAAI Conference on Artificial Intelligence **34**(05), 8968–8975 (Apr 2020). https://doi.org/10.1609/aaai.v34i05.6428, https://ojs.aaai.org/index.php/AAAI/article/view/6428
46. Topsakal, O., Akinci, T.C.: Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In: Proceedings of the International Conference on Applied Engineering and Natural Sciences, Konya, Turkey. pp. 10–12 (2023)
47. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
48. Wang, C., Yu, H., Zhang, Y.: Rfid: Towards rational fusion-in-decoder for open-domain question answering. arXiv preprint arXiv:2305.17041 (2023)
49. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652 (2021)
50. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=gEZrGCozdqR

51. Xiao, D., Li, Y.K., Zhang, H., Sun, Y., Tian, H., Wu, H., Wang, H.: ERNIE-gram: Pre-training with explicitly n-gram masked language modeling for natural language understanding. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1702–1715. Association for Computational Linguistics, Online (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.136, https://aclanthology.org/2021.naacl-main.136
52. Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., Lin, J.: End-to-end open-domain question answering with bertserini. In: Ammar, W., Louis, A., Mostafazadeh, N. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations. pp. 72–77. Association for Computational Linguistics (2019). https://doi.org/10.18653/V1/N19-4013, https://doi.org/10.18653/v1/n19-4013
53. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600 (2018)
54. Yasunaga, M., Leskovec, J., Liang, P.: LinkBERT: Pretraining language models with document links. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 8003–8016. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.551, https://aclanthology.org/2022.acl-long.551
55. Yu, Z., Xiong, C., Yu, S., Liu, Z.: Augmentation-adapted retriever improves generalization of language models as generic plug-in. arXiv preprint arXiv:2305.17331 (2023)
56. Zhong, Z., Lei, T., Chen, D.: Training language models with memory augmentation. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. pp. 5657–5673. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.EMNLP-MAIN.382, https://doi.org/10.18653/v1/2022.emnlp-main.382
57. Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., Chua, T.S.: Retrieving and reading: A comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774 (2021)