KBM: Delineating Knowledge Boundary for Adaptive Retrieval in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) often struggle with dynamically changing knowledge and handling unknown static information. Retrieval-Augmented Generation (RAG) is employed to tackle these challenges and has a significant impact on improving LLM performance. In fact, we find that not all questions need to trigger RAG. By retrieving parts of knowledge unknown to the LLM and allowing the LLM to answer the rest, we can effectively reduce both time and computational costs. In our work, we propose a **K**nowledge **B**oundary **M**odel (KBM) to express the known/unknown of a given question, and to determine whether a RAG needs to be triggered. Experiments conducted on 11 English and Chinese datasets illustrate that the KBM effectively delineates the knowledge boundary, significantly decreasing the proportion of retrievals required for optimal end-to-end performance. Furthermore, we evaluate the effectiveness of KBM in three complex scenarios: dynamic knowledge, long-tail static knowledge, and multi-hop problems, as well as its functionality as an external LLM plug-in.

1 Introduction

As Large Language Models (LLMs) evolve, their real-world applications expand, yet they often struggle with dynamically changing and unknown static knowledge, leading to inaccuracies or *hallucinations* (Rawte et al., 2023). Retrieval-Augmented Generation (RAG) effectively addresses challenges by retrieving relevant external information in real time, enhancing LLMs' accuracy. However, it also incurs costs, such as increased retrieval requests and longer response times, leading to the crucial question: When is retrieval truly necessary?

A natural criterion for determining the need for retrieval is the confidence level of the LLM. Specifically, the known knowledge of the model does not require retrieval, while uncertain parts can benefit from this process. As illustrated in Figure 1 (top),

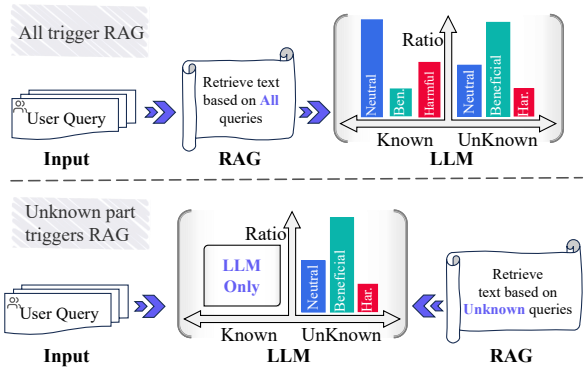


Figure 1: Illustration of the impact of different RAG triggering methods on LLM. Beneficial: RAG effectively solves this question. Neutral: RAG has no effect on the LLM. Harmful: RAG reduces the LLM's effectiveness.

using RAG for all questions (All RAG) can result in an overall impact that is neutral or even harmful, as the increased retrieval calls may not justify the benefits. In contrast, Figure 1 (bottom) demonstrates that when focusing solely on the uncertain part through RAG, the benefits are substantial, while reducing the frequency of unnecessary retrieval calls. The key to implementing this solution is how to enable LLM express confidence. A simple and effective solution is to use prompts. Unfortunately, over-/under-confidence will affect the expression of confidence (Xiong et al., 2024).

Previous studies on whether RAG is required for LLM can be divided mainly into two approaches. The first focuses on the *question* itself, with methods like Self-RAG (Asai et al., 2023) instructing models such as GPT-4 (Achiam et al., 2023) to assess whether retrieving external documents (e.g., Wikipedia) can produce better responses. Although this approach can identify questions that require real-time information, it remains model-agnostic and struggles to determine whether an LLM has mastered specific knowledge. The second approach evaluates both *questions* and *model responses* to

determine if an LLM can answer a question, generating data by sampling multiple model responses and using manual labels for evaluation (Ren et al., 2024; Yin et al., 2023; Kadavath et al., 2022). However, this method is labor intensive and relies heavily on manual labeling, which can create biases and lead to increased training costs. Additionally, the LLM's "unknown" expression has not been integrated with RAG triggering to assess its impact on end-to-end performance and retrieval ratio.

In this work, we propose the Knowledge Boundary Model (KBM) and two solutions for SFT based on *accuracy* and *certainty* in generating soft labels. The first method evaluates sampled QA accuracy and sets a threshold to classify questions as known or unknown. The second approach is based on certainty, focusing on modeling the entropy values of multiple sampled responses to establish thresholds, distinguishing between known and unknown responses through consistency, without requiring a golden answer. By defining different accuracy and certainty thresholds, we can generate data with soft labels, which is then used to fine-tune the KBM to determine whether the LLM considers a question unknown, thus indicating if retrieval is necessary.

We evaluate 11 English and Chinese QA datasets, demonstrating that KBM effectively measures knowledge mastery in LLMs. On average in these datasets, the certainty-based approach also reduces the number of retrievals by 13. 5% compared with ALL RAG, resulting in a slightly higher performance of 0.1%. Similarly, the accuracy-based approach reduces retrievals by 32.5% on average, with only a 0.39% performance drop. MFurther analysis confirms the effectiveness of KBM in open domains, including dynamic knowledge, long-tail static knowledge, and multi-hop scenarios.

Our contributions are summarized as follows:

- To our knowledge, KBM is the first study to reduce the RAG trigger ratio while maintaining LLM performance, thereby enhancing RAG's efficiency in QA tasks and reducing costs.
- For the technique contribution, we propose two methods for generating soft labels based on accuracy and certainty, allowing LLMs to express "unknown" or "known".
- KBM is validated on 11 datasets, demonstrating comparable effectiveness with All RAG and a reduced retrieval ratio, and it performs well in three complex scenarios.

2 Preliminaries: LLM Knowledge Boundary and RAG Analysis

This section examines the impact of RAG on LLM performance. We evaluate the knowledge boundaries of LLMs with different parameter sizes and analyze RAG's effects on QA tasks, categorizing questions into three types based on its influence on LLM responses. Lastly, we introduce sampling methods based on LLM accuracy and certainty to further simulate knowledge boundary.

2.1 How does RAG Affect the Accuracy of LLM Response?

We evaluate how RAG affects the performance of LLMs with different parameter sizes. Our results show that while LLMs vary in their QA abilities and knowledge limits, their use of retrieved information is fairly consistent. We use three configurations: LLM Only generates responses from the LLM alone; ALL RAG enhances Naive RAG by adding the top 10 blocks retrieved from Google as context; and MASK RAG substitutes the correct answers in RAG with MASK, providing this altered data as context for the LLM.

We focus on the Qwen1.5 models (4B, 7B, 14B, 32B) (Bai et al., 2023) alongside the Qwen2.0 72B model (Yang et al., 2024), using evaluation datasets for short question answering and reading comprehension tasks, including WebQA (Chang et al., 2022), SogouQA[1], and SQuAD1.5-zh[2]. Our results show that LLM performance improves with larger parameter sizes across all three datasets, leading to a gradual differentiation in their QA capabilities. Notably, with Naive RAG, all models, especially those with 14B-72B parameters, effectively utilize contextual information. RAG has a more significant impact on smaller LLMs, but the maximum performance gain is similar across different model sizes. For example, on the WebQA dataset, the accuracy difference between the 7B and 72B models using RAG is just 3.05%, compared to 28.87% without RAG. Interestingly, MASK RAG seems to reduce the benefits of RAG, potentially harming performance on simpler datasets, as noisy information negatively affects smaller models more. For detailed results, see Appendix §A Figure 9.

These findings suggest that different LLMs possess varying knowledge boundary in question answering and demonstrate distinct retrieval strate-

[1]https://github.com/sherlcok314159/ChineseMRC-Data
[2]https://github.com/pluto-junzeng/ChineseSquad

gies. Although all models exhibit strong capabilities in leveraging context, they show varying degrees of resilience to noise interference.

2.2 Three Impacts of RAG on LLM

To accurately assess the impact of RAG on LLMs, we categorize its effects into three aspects:

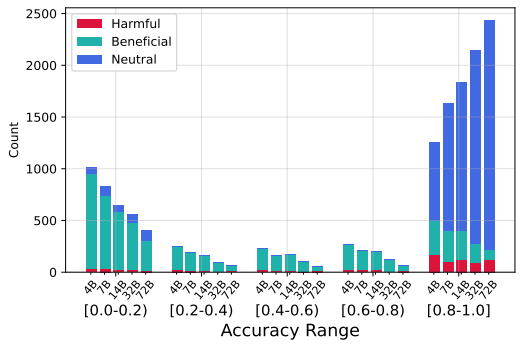
- Beneficial: RAG effectively solves this question.
- Neutral: RAG has no effect on the LLM.
- Harmful: RAG reduces the LLM's effectiveness.

To isolate the impact of other modules, we use a simplified RAG pipeline for analysis, retrieving the top 10 blocks from Google Open Search to input into the LLM. To improve robustness, we sample each question 30 times. Inspired by Kadavath et al. (2022), we generate $I = 30$ answer samples at temperature $T = 1$. For a given question Q, if the model samples 20 correct answers and 10 incorrect ones, we construct an average accuracy of knowledge for that question based on these samples, resulting in a single data point $(Q, M_{pred} = \frac{20}{30})$. Unlike the approach in (Kadavath et al., 2022), which includes 20 copies of (Q, M=1) and 10 copies of (Q, M=0) our method accurately represents the model's understanding and misconceptions while significantly reducing the size of the training and test datasets by a factor of I. Thus, we approximate the model's soft labels for knowledge using hard labels derived from a diverse set of QA data points.

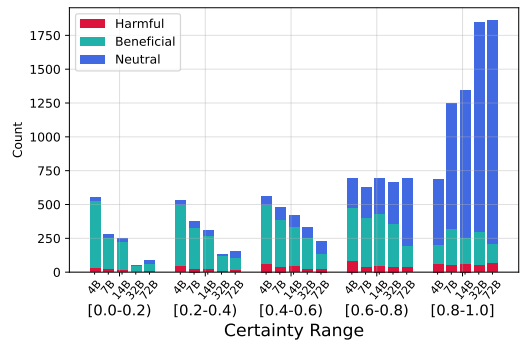
However, this method becomes challenging in the absence of gold-standard answers. To address this issue, we simulate the effect based on the certainty of the generated responses. Specifically, we compute the entropy distribution of words and phrases from the 30 generated answers to establish the model's level of certainty. Let k represent the number of distinct answer types, denoted as $K_1, K_2, ..., K_k$. The probability of each answer type occurring is represented as $P_1, P_2, ..., P_k$, satisfying the normalization condition: $\sum_{i=1}^{k} P_i = 1$. Using this probability distribution, we quantify the model's certainty through entropy. First, we define entropy H as:

$$H = -\sum_{i=1}^{k} P_i \log_2(P_i), \tag{1}$$

and the maximum possible entropy $H_{max} = \log_2(k)$. The model's certainty $C(Q)$ is calculated



(a) Category distribution based on **Accuracy** interval.



(b) Category distribution based on **Certainty** interval.

Figure 2: Category distribution across various indicators: (a) Accuracy and (b) Certainty intervals.

by normalizing the entropy:

$$C(Q) = 1 - \frac{H}{H_{max}}, \tag{2}$$

where $H/H_{max} \in [0, 1]$ is the normalized entropy.

We classify the effects of RAG based on the three categories and two indicators as follows: an effect is deemed **Beneficial** if the indicator increases after incorporating RAG compared to the LLM Only scenario; **Neutral** if the indicator remains unchanged and aligns with the value from the LLM Only condition; and **Harmful** if the indicator decreases following the addition of RAG, falling below the value observed in the LLM Only scenario. Using these categories, we analyze the class distribution of LLMs across accuracy intervals on the WebQA dataset, as shown in Figure 2(a). The figure indicates that the [0-0.8) interval has the highest proportion of Beneficial cases, while Neutral and Harmful cases peak in the [0.8-1.0] range. This suggests a cost-benefit relationship: higher accuracy reflects greater confidence in answers, reducing the advantages of RAG, especially in the [0.8-1.0] range, where harmful cases also concentrate. High certainty reflects high confidence, while low cer-

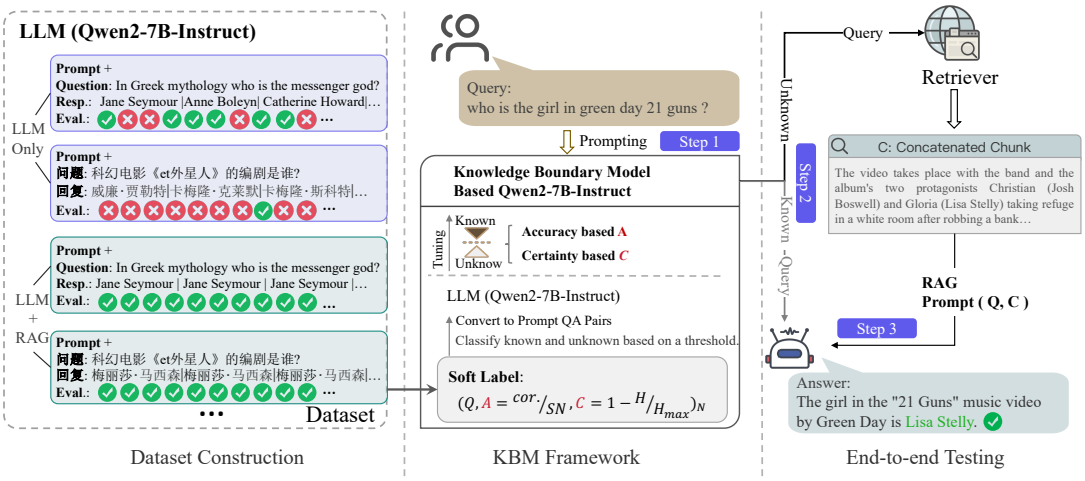


Figure 3: Illustration of the workflow for data generation, model training, and inference processes.

tainty indicates doubt. The certainty metric considers factors such as chance and model uncertainty (Hu et al., 2023). We find a Pearson correlation coefficient of 0.64 between accuracy and certainty, as shown in Figure 2. Certainty also has a similar category distribution to accuracy, indicating a positive correlation.

In the high accuracy range (above 0.8), we first note that many neutral examples unnecessarily trigger RAG. Additionally, harmful examples can lead to a sharp accuracy drop, potentially from 1.0 to 0.0, while beneficial examples typically increase accuracy from 0.8 to a maximum of 1.0. This leads us to propose a threshold to differentiate between known and unknown data. Data below this threshold is marked as unknown and triggers RAG, while data above it is considered known, allowing the LLM to respond independently.

3 Can LLM Express Know/Unknown?

In this section, we generate known and unknown soft labels based on accuracy and certainty for training the KBM, and we introduce the baselines and datasets used for evaluation.

3.1 Methods

To construct training data for the KBM, we generate soft labels using the sampling method detailed in Section §2.2. As shown in Figure 3, each query is assessed under two configurations: LLM Only and LLM + RAG, providing scores for *accuracy* and *certainty*. A threshold of $\tau = 0.9$ is set to classify data as *known* or *unknown*. The data is then formatted into QA pairs for fine-tuning the Qwen2-7B model. For additional setup details, refer to Appendix §B.

During the inference phase of the KBM, the process unfolds in three steps: At **step 1**, after the user enters a query, the system packages this query through a prompt and sends it to the KBM to assess accuracy and certainty. At **step 2**, if the KBM judges the query as known, it forwards the original query directly to the answer generation model (Qwen2-7B instruct) to generate a response. If the query is judged as unknown, the process proceeds to step 3. At **step 3**, the system performs an open-domain retrieval using Google, based on the user's original query. The retrieved chunks of information are then spliced together. The original query and the connected chunks are combined using RAG prompt and subsequently fed into the answer generation model to produce the final response.

3.2 Experiment Setting

Metrics: For the English test data, we use Exact Match (EM) as the metric, while Accuracy is applied to the other datasets, represented as A/E. Additionally, we consider the retrieval ratio (Rat.) as a crucial metric.

Baselines: We establish a baseline using the following methods: (1) *Prompt*: we use prompts from Qwen2-7B (Yang et al., 2024), Llama3-70B (AI@Meta, 2024), Qwen2-72B (Yang et al., 2024), and GPT-4o (OpenAI et al., 2024) to determine whether the current query requires retrieval. Each model is presented with the same retrieval query. (2) *Random*: This method serves as a dynamic benchmark. By analyzing the ratio of each baseline method that triggers RAG, a query from the test set is randomly selected to initiate the RAG evalu-

		NQ	TriQA	OBQA	MMLU	SQD	MLEC	WebQA	SogQA	zh_SQD	CMMLU	CSQA	Avg.
Naive *(Qwen2-7B-Instruct based + Google)*													
LLM	A/E	35.58	50.79	82.72	66.45	34.02	74.25	70.06	52.53	22.72	74.23	24.13	53.41
RAG	A/E	61.63	84.60	81.48	70.80	89.83	68.40	92.54	83.01	83.66	77.05	77.80	79.16
Qwen2-7B-Instruct based + Google													
	A/E	59.92	79.08	81.96	67.30	78.55	72.80	89.75	78.13	78.74	77.97	75.41	76.33
Prompt	Rnd.	57.91	75.62	82.12	67.70	75.30	72.01	87.01	76.75	77.82	72.15	75.14	74.50
	Rat.	86.0%	73.6%	27.8%	27.2%	75.6%	44.7%	77.5%	78.9%	90.3%	49.7%	94.1%	65.9%
Self-RAG *(Llama-2 based + Google)*													
	A/E	31.07	69.82	74.80	47.93	62.92	29.48	52.96	39.83	29.47	33.94	18.03	44.57
- 7B	Rnd.	35.79	69.49	76.40	49.38	62.27	30.72	54.02	39.83	30.33	34.63	19.17	45.64
	Rat.	22.7%	55.9%	10.8%	29.9%	77.0%	10.4%	28.3%	28.5%	37.0%	3.4%	23.2%	29.7%
KBM *(Qwen2-7B-Instruct based + Google)*													
	A/E	60.09	81.39	**82.64**	70.00	89.13	**73.28**	90.84	81.59	82.66	**78.32**	76.57	78.77
- Acc.	Rnd.	51.20	74.42	81.92	68.59	83.14	72.66	81.19	74.17	81.29	77.73	74.87	74.65
	Rat.	86.2%	70.3%	26.2%	70.0%	87.2%	30.2%	56.8%	75.2%	96.0%	50.0%	94.3%	67.5%
	A/E	**61.27**	**83.32**	82.32	**70.45**	**89.38**	71.69	**92.12**	**82.89**	**83.45**	77.12	**77.38**	**79.22**
- Cer.	Rnd.	55.16	80.66	82.88	69.14	88.61	69.46	90.71	81.06	83.31	77.06	76.85	77.72
	Rat.	97.5%	87.7%	57.8%	65.3%	97.3%	69.0%	89.7%	94.6%	99.3%	95.3%	97.8%	86.5%

Table 1: Comparison of performance and trigger ratio (Rat. %) metrics between KBM and baseline models. The English dataset utilizes EM as the performance metric, while the Chinese dataset employs accuracy as the evaluation metric. Each method is compared with the random RAG trigger score (Rnd.) at the same trigger ratio.

ation. This approach effectively assesses whether the model has genuinely identified the unknown query or merely "guessed" it. Among related approaches, we select *Self-RAG* (Llama-2 7B based) (Asai et al., 2023) as baselines for our analysis. We use Google as the retriever for all methods, selecting the first 10 snippets as contextual information. The experiment focuses on end-to-end effectiveness, the reduction in retrieval ratio, and the differences between the baseline methods and the random approach.

Datasets: KBM's training and test data encompass a variety of task types, including short answer questions, multiple choice questions, reading comprehension, and multi-hop questions. We utilize TriviaQA (TriQA) (Joshi et al., 2017), WebQA (Chang et al., 2022), and a combination of MMLU (Hendrycks et al., 2021), MLEC (Li et al., 2021), and XieZhi (Gu et al., 2024) training sets to train the KBM model. We categorize the test sets into two groups: **In-Domain**, which includes TriviaQA (Joshi et al., 2017), WebQA (Chang et al., 2022), MLEC (Li et al., 2021), and MMLU (Hendrycks et al., 2021), and **Out-Of-Domain**, which comprises NaturalQA(NQ) (Kwiatkowski et al., 2019), Open BookQA (OBQA) (Mihaylov et al., 2018), SQuAD (SQD) (Rajpurkar et al., 2018), FreshQA (Vu et al., 2023), SogouQA (SogQA) [3], the Chinese SQuAD (zh_SQD) [4], CMMLU (Li et al., 2024a), and Chinese SimpleQA (CSQA) (He et al., 2024). For details on the datasets, see the Appendix §B.2.

[3]https://github.com/sherlcok314159/ChineseMRC-Data
[4]https://github.com/pluto-junzeng/ChineseSquad

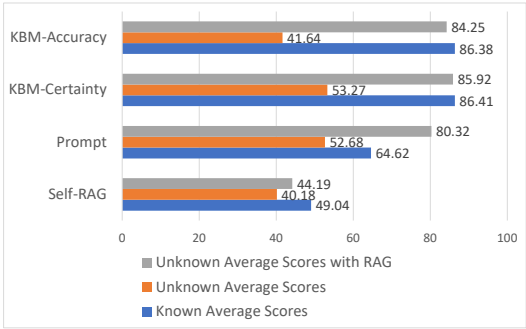


Figure 4: The performance of LLM in the case of KBM model being judged as known/unknown (0-100). The higher the average score of known, the better.

4 Experiments

In this section, we design extensive experiments to verify the effectiveness of KBM.

4.1 End-to-End Evaluation

We evaluate the end-to-end performance of KBM across 11 test sets, showing that both accuracy and certainty reduce the RAG trigger ratio while enhancing the LLM's ability to answer questions. The results are presented in Table 1. On average, the KBM accuracy-based method outperformed the LLM Only method by 25.4%, narrowing the gap with All RAG to 0.4%, while decreasing the RAG triggering ratio by 32.5%, which is 4.1% higher than the Random method, indicating a greater beneficial density. Similarly, the KBM certainty-based method improved performance by 25.8% compared to the LLM Only method, achieving a performance

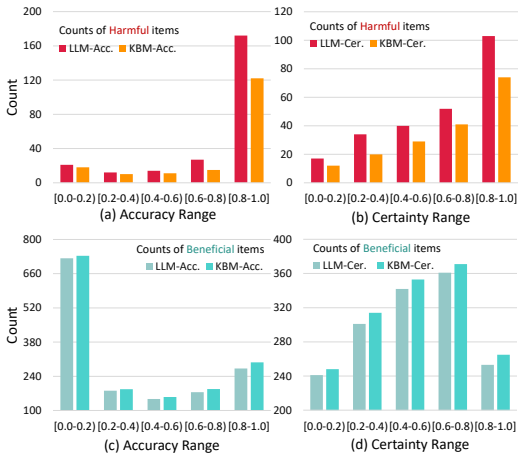


Figure 5: Illustration of the KBM method's effects: the top shows a decrease in harmful counts, while the bottom indicates an increase in beneficial counts.

that is 0.1% higher than All RAG, and reduced the retrieval ratio by 13.5%. Compared with the Prompt and Self-RAG methods, KBM effectively reduces the RAG triggering proportion while maintaining high performance.

Specifically, our analysis reveals that for short answer and QA test sets (e.g., NQ, TriviaQA, WebQA, SogouQA, and CSQA), both methods outperform the random baseline, with accuracy achieving particularly notable results. In reading comprehension tasks like SQA and zh_SQD, where background information is essential for accurate answers, KBM exhibits a higher trigger ratio. All methods demonstrate strong performance and a high trigger ratio on FreshQA, indicating that KBM and LLM effectively capture queries with temporal features. However, in the context of multiple-choice questions found in CMMLU, MMLU, and OBQA, our method shows only marginal improvements over the random baseline. We guess that this limited enhancement arises from the challenges of sourcing relevant information for multiple-choice formats. Moreover, the diversity of training data formats, in relation to data type and task domain, is crucial for the effectiveness of KBM.

Known/Unknown Expression Analysis. KBM enables the model to effectively distinguish between known and unknown information, thereby optimizing performance and reducing costs. We analyze the mean scores across all test sets categorized as known or unknown by KBM, as illustrated in Figure 4. We can see that without the assistance of RAG (represented by the orange bar), the LLMs struggle to handle unknown questions. In contrast, the two KBM methods achieve the most significant improvements compared to other approaches. Specifically, the prompt-based method shows 'confusion' about the knowledge boundary because there is a notable gap between Known Aeverage Scores and Unknown Average Scores with RAG. For improvements in accuracy and certainty at finer intervals, see the Appendix §A.1.

	fast-changing		slow-changing		never-changing	
	Ratio	Acc.	Ratio	Acc.	Ratio	Acc.
Naive *(Qwen2-7B-Instruct based + Google)*						
LLM	0%	29.7	0%	29.8	0%	40.2
RAG	100%	50.4	100%	60.2	100%	66.6
Prompt	93.8%	51.5	86.8%	57.8	81.8%	64.1
Self-RAG(7B)	56.2%	37.4	56.3%	49.4	55.1%	56.0
KBM-Acc.	94.6%	51.6	92.3%	59.0	75.9%	62.7
KBM-Cer.	98.5%	**52.0**	95.6%	**60.2**	89.4%	**64.6**

Table 2: Results of triggering RAG with various levels of dynamic knowledge.

Impact Analysis of KBM on RAG. In Section §2 Figure 2, we discuss the three impacts of RAG on LLM. To investigate the changes in the number of harmful and beneficial questions, we employ KBM instead of RAG and find that KBM reduces the number of harmful questions while increasing the number of beneficial ones. Specifically, as shown in Figures 5(a) and 5(b), using WebQA as an example, we find that compared with RAG, the KBM-based approach decreases the number of harmful questions across all five intervals. This reduction is particularly evident in the high accuracy interval, indicating that we mitigate the harmful impact associated with RAG in areas close to what is known. Similarly, as illustrated in Figures 5(c) and 5(d), KBM increases the number of beneficial questions in each interval, showing that it enhances system performance by effectively triggering RAG. Moreover, we conduct a comprehensive efficiency analysis, the results of which are presented in §A.3.

4.2 Analysis of Complex Scenarios

A multitude of complex scenarios necessitate the frequent invocation of RAG. In light of this, we conducted tests and focused on the **ratio** of KBM in the following scenarios.

Dynamic Knowledge. We demonstrate that KBM effectively identifies questions with answers that change over time. Specifically, we classify the temporal changes in answers found in the FreshQA dataset into three distinct categories: fast-changing, slow-changing, and never-changing, based on the frequency of these changes. In open domains, vari-

	en-FreshQA(≥2hop)		HotpotQA	
	Ratio	Acc.	Ratio	Acc.
LLM	0%	26.1	100%	30.5
RAG	100%	48.7	100%	51.7
Prompt	91.3%	48.3	93.7%	51.2
Self-RAG(7B)	64.3%	47.5	65.0%	38.6
KBM-Acc.	96.5%	**48.9**	91.2%	50.9
KBM-Cer.	99.1%	48.7	94.3%	**51.4**

Table 3: Results of multi-hop triggering RAG.

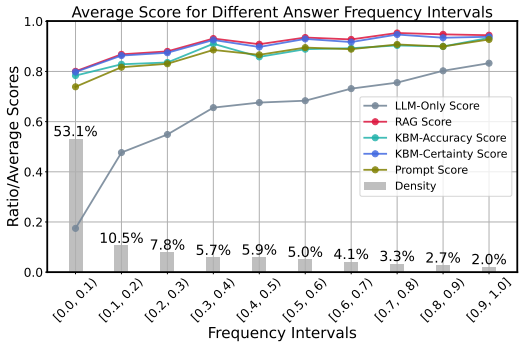


Figure 6: Comparison of performance changes under different knowledge frequencies.

		Gpt-4o		Qwen2-72B		Llama3-70B	
		WebQA	NQ	WebQA	NQ	WebQA	NQ
LLM	A/E	64.3	55.6	83.7	42.7	48.7	47.2
RAG	A/E	90.0	60.1	91.9	59.4	90.9	60.2
Prompt	A/E	69.5	55.7	87.9	54.7	59.7	59.2
	Rnd.	60.8	56.6	85.6	52.6	55.3	58.6
	Rat.	14%	18%	22%	51%	16%	47%
Self-RAG (7B)	A/E	72.6	56.7	86.5	45.7	60.9	49.6
	Rnd.	71.6	57.1	86.0	42.7	48.7	47.2
	Rat.	28%	23%	28%	23%	28%	23%
KBM-Acc.	A/E	85.7	60.6	91.4	58.5	77.9	59.6
	Rnd.	78.5	60.4	88.2	56.5	72.6	58.6
	Rat.	57%	86%	57%	86%	57%	86%
KBM-Cer.	A/E	**89.5**	**60.7**	**91.8**	**59.3**	**88.2**	**60.2**
	Rnd.	87.8	60.5	91.3	58.9	86.7	59.7
	Rat.	90%	98%	90%	98%	90%	98%

Table 4: End-to-end result with KBM as a plug-in for various LLMs' retrieval judgment modules.

ations in answers often indicate the need for knowledge updates, necessitating the integration of external information into the LLM. As shown in Table 2, the end-to-end performance without RAG for both fast-changing and slow-changing categories is suboptimal, highlighting a reliance on external knowledge. The high RAG trigger rates of KBM for these categories suggest that it effectively captures evolving answers. Conversely, the lower retrieval rate for the never-changing category implies that some knowledge is effectively embedded within the LLM. This finding underscores KBM's sensitivity in identifying questions with temporally correlated answers and highlights its role in enhancing dynamic knowledge adaptation.

Multi-Hop. A crucial aspect of our analysis is the ability of KBM to detect complex queries that necessitate multi-hop knowledge. Multi-hop questions comprise intricate knowledge components, requiring adjustments to the LLM. In these scenarios, KBM identifies the complexity of the queries and effectively employs RAG. We tested queries involving two or more hops from the FreshQA(≥2-hop) and HotpotQA test sets, with results presented in Table 3. KBM demonstrates higher retrieval rates for multi-hop questions, indicating that it reveals more unknowns when addressing these complexities. As a result, the trigger ratio is higher than the average ratio for the task. While the overall end-to-end improvement is modest, this limitation stems from the need for further optimization of the RAG pipeline. We employ KBM to assess its ability to detect complex problems and trigger RAG, rather than to break them down and resolve them.

Long-tail Static Knowledge. Long-tail knowledge has consistently posed challenges for the learning process of LLMs (Kandpal et al., 2023). We investigate the capacity of KBM to capture low-frequency long-tail knowledge across various question sets, confirming its effectiveness across all frequency ranges. Specifically, we combine test data from WebQA, SogQA, and zh_SQD. Utilizing the gold answers from these datasets, we conduct vector retrieval within our Chinese database to differentiate knowledge based on its frequency. As illustrated in Figure 6, the LLM Only approach demonstrates reduced accuracy for low-frequency knowledge answers while performing better for high-frequency knowledge. However, integrating KBM with a prompt-based retrieval mechanism significantly enhances the model's performance for long-tail low-frequency knowledge. Notably, the certainty-based method yields the most substantial improvement, followed by the accuracy-based approach. These findings indicate that KBM effectively detects low-frequency long-tail knowledge and boosts overall performance by triggering RAG.

4.3 Use as a Plug-in

To assess the effectiveness of the KBM model in triggering RAG, we apply it as a plug-in to GPT-4o, Llama3-70B, and Qwen2-72B. We evaluate these models using the WebQA and NQ test sets. The

results are summarized in Table 4. Our findings indicate that KBM enhances the performance of these LLMs, although it does not achieve the comprehensive improvements provided by ALL RAG. This is consistent with our previous analysis showing that each LLM has a unique knowledge boundary, complicating the representation of knowledge boundary across all LLMs with a single model. The varying knowledge boundary of LLMs significantly influence the enhancement effects observed with knowledge boundary models (KBM). For example, the KBM-certainty judgment method shows that Qwen2-72B improves its performance from 83.7% to 91.8%, resulting in an increase of 8.1%, while Llama3-70B experiences a much larger gain, rising from 48.7% to 88.2%, equating to a 39.5% improvement. This illustrates that the initial knowledge capacity and boundary of an LLM can lead to divergent levels of enhancement, when interfacing with KBMs.

Additionally, RAG effectively enhances the response quality of LLMs, leading to a relatively stable upper limit for improvements across models. However, the performance gains differ due to the varying distributions of known and unknown knowledge in KBMs versus general models. For instance, although Llama3-70B demonstrates a notable retrieval rate of 90%, its upper limit is lower than that of Qwen2-72B, which struggles to reach the ALL RAG score despite the same retrieval performance in the WebQA dataset. These trends are echoed in English QA tests, where significant performance improvements are observed when the English proficiency of the KBM ontology model inferior to that of the generative LLM. This suggests that aligning the knowledge boundary with the capabilities of the LLM is essential for optimizing performance.

5 Related Work

Large Language Model Knowledge Exploration. The exploration of knowledge boundary in LLMs attracts significant attention. Kadavath et al. (2022) examines the self-evaluation capabilities of LLMs, showing that larger models enhance their calibration by initially proposing answers and then evaluating their validity. Ren et al. (2024) studies LLMs' perception of factual knowledge boundary and finds that they often display blind confidence in their abilities. Yin et al. (2023) focuses on self-awareness, demonstrating that while LLMs can identify some unanswered questions, substantial discrepancies still exist, affecting their uncertainty detection. Chen et al. (2024) introduces COKE, an unsupervised method for teaching models to articulate their knowledge limits through internal signals, yielding improved outcomes across various datasets. Kang et al. (2024) points out that LLMs often default to examples in training data when facing unfamiliar queries. Li et al. (2024b) explores hallucinations related to insufficient prompt context, showing that models frequently fail to recognize inadequate information.

Retrieval-Augmented Generation. RAG enhances LLMs by integrating retrieved text passages, significantly improving performance in knowledge-intensive tasks. A key focus is optimizing the timing and strategy of retrieval. Asai et al. (2023) introduce SELF-RAG, a method that trains LLMs to retrieve information, generate content, and evaluate their outputs using reflection tokens. This method enables the customization of model behavior, demonstrating significant performance improvements over standard RAG approaches. Jeong et al. (2024) proposes Adaptive-RAG, which adjusts query handling based on complexity. Wu et al. (2024) explores how LLMs process erroneous retrieved content. By creating a dataset to assess model responses to incorrect information, the study reveals insights into how models correct their outputs or may perpetuate errors. Cuconasu et al. (2024) conducts a comprehensive study on the retriever's function in RAG.

6 Conclusion

In this paper, we propose KBM to address the limitations of LLMs in managing dynamic knowledge and unknown static information. KBM judges whether to trigger RAG by indicating the known and unknown of a question, preserving RAG's advantages while reducing computational costs. Validated on 11 datasets, KBM matches the performance of All RAG while reducing the retrieval ratio, enhancing efficiency in QA tasks without compromising other instruction fine-tuning performance. We also confirm KBM's robustness in complex scenarios, such as dynamic and long-tail knowledge and multi-hop problems, highlighting its effectiveness as an external LLM plug-in. Our contributions include methods for generating soft labels based on accuracy and certainty, enabling LLMs to better express their knowledge status.

Limitations

Our work is the first attempt to explore how to reduce the proportion of retrievals that trigger RAG while maintaining high performance. Despite the reduction in the proportion of triggers, our study has some limitations that may guide future efforts.

First, we propose a method based on accuracy and certainty sampling to generate soft labels for training KBM, demonstrating that LLMs learn to express unknowns in different tasks without requiring a large amount of data. However, in real-world scenarios, it is essential to include more diverse QA format data for better generalization, such as reasoning and content generation. Additionally, converting data from various question-answer formats into the format of QA pairs enhances the performance of KBM.

Secondly, while we enable LLMs to express unknowns and trigger RAG, there remains a relatively low probability that an unknown question posed by LLM can be resolved by RAG, particularly when considering post-retrieval operations such as changing the retriever, sorting, and denoising.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, et al. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. *Preprint*, arXiv:2109.00590.

Lida Chen, Zujie Liang, Xintao Wang, Jiaqing Liang, Yanghua Xiao, Feng Wei, Jinglei Chen, Zhenghong Hao, Bing Han, and Wei Wang. 2024. Teaching large language models to express knowledge boundary from their own signals. *Preprint*, arXiv:2406.10881.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Yixin Zhu, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Weijie Wu, et al. 2024. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18099–18107.

Yancheng He, Shilong Li, Jiaheng Liu, Yingshui Tan, Weixun Wang, Hui Huang, Xingyuan Bu, Hangyu Guo, Chengwei Hu, Boren Zheng, Zhuoran Lin, Xuepeng Liu, Dekai Sun, Shirong Lin, Zhicheng Zheng, Xiaoyong Zhu, Wenbo Su, and Bo Zheng. 2024. Chinese simpleqa: A chinese factuality evaluation for large language models. *Preprint*, arXiv:2411.07140.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *Preprint*, arXiv:2306.04459.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. 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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, et al. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *Preprint*, arXiv:2403.05612.

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.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024a. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Jiaqi Li, Yixuan Tang, and Yi Yang. 2024b. Know the unknown: An uncertainty-sensitive method for llm instruction tuning. *Preprint*, arXiv:2406.10099.

Jing Li, Shangping Zhong, and Kaizhi Chen. 2021. MLEC-QA: A Chinese Multi-Choice Biomedical Question Answering Dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8862–8874, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2024. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *Preprint*, arXiv:2307.11019.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2023. Freshllms: Refreshing large language models with search engine augmentation. *Preprint*, arXiv:2310.03214.

Kevin Wu, Eric Wu, and James Zou. 2024. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. *Preprint*, arXiv:2404.10198.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

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.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *Preprint*, arXiv:2305.18153.

Figure 7: Case study of the KBM and effects of combining KBM with SFT data in a QA scenario.

A Additional Experimental Analysis

A.1 Analysis of KBM in Sub-intervals

We divided the accuracy interval into steps of 0.2 on the Chinese WebQA test set to investigate the improvements of KBM compared to RAG. As illustrated in Figure 8, our findings indicate that the KBM method, based on Accuracy and Certainty, demonstrates significant enhancements across several intervals, particularly in the lower accuracy ranges. For instance, when the LLM-only accuracy is at 0.0, KBM-Certainty shows an increase of 79.4%, while KBM-Accuracy exhibits a rise of 74.2%. Notably, when compared to All RAG, KBM achieves performance levels that are comparable to RAG in multiple intervals, especially within the Certainty-based approach.

A.2 Case Study

As illustrated in Figure 7, we showcase how KBM functions within a QA context. For questions (1) and (4), if KBM determines that the answer is Known, the LLM can provide the correct response directly, eliminating the need to search for external resources. In the case of question (2), which

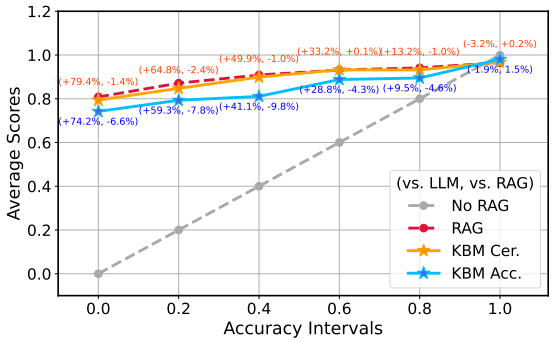


Figure 8: Illustration of the relative improvement of KBM for different accuracy ranges of LLM Only.

the LLM cannot answer or chooses not to address, KBM can preemptively classify it as unknown. This classification enables subsequent actions to penalize the search tool, prompting it to retrieve relevant textual materials that can then be provided to the LLM as context to aid in formulating an answer. For questions (3) and (5), direct use of the LLM to respond may lead to hallucinations, resulting in the output of incorrect information. In these instances, KBM can classify the question as unknown and activate RAG to produce a more reliable answer.

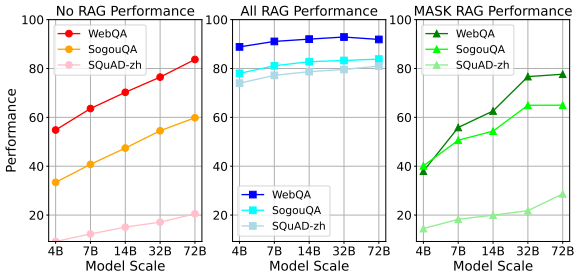


Figure 9: Illustration of the impact of RAG on LLM performance.

	A/E	Search API Call Rate	Token consumption	time (ms)
All RAG	79.16	100.00%	593	2498
KBM-Acc.	78.77	**67.50%**	**421**	**2231**
KBM-Cer.	**79.22**	86.50%	517	2384

Table 5: Efficiency analysis across 11 datasets.

A.3 Efficiency analysis

We evaluate our method across 11 datasets and compare it to the All RAG approach. The findings appear in Table 5. For KBM-Acc, the API calls reduce by **32.5%**; tokens input to LLM decrease from 593 to 421; and the total computation time reduces from 2498 ms to 2231 ms. These results indicate that KBM-Acc offers significant advantages in both computational and invocation costs compared to All RAG. On the other hand, when comparing with All RAG, our KBM-Cer reduces all three metrics, namely API calls, token consumption, and time, to varying degrees. Importantly, KBM-Cer not only controls costs but also **surpasses All RAG** in overall performance metrics.

B Implementation Details

B.1 Parameter Settings

We employ four Nvidia A100 GPUs, each with 80GB of memory, to train our KBM models. To maintain high end-to-end performance, we set the threshold for both Accuracy and Certainty at 0.9. Each model undergoes training for three epochs, utilizing a batch size of four. The peak learning rate is set to 1e-5, with a warmup ratio of 2% and cosine decay for the learning rate. To accommodate memory limitations, we restrict the maximum token length to 1580 for the 7B model and 1524 for the 13B model. For multi-GPU distributed training, we utilize Deepspeed stage 2 (Rajbhandari et al., 2020) while enabling Bfloat16 precision. Inference on the trained models is conducted using a single Nvidia Tesla V100 GPU with 32GB of memory.

B.2 Datasets

English Datasets. NaturalQA (Kwiatkowski et al., 2019): This QA dataset, curated by Google, consists of real-world questions derived from natural retrieval queries. TriviaQA (Joshi et al., 2017): This dataset is based on encyclopedic content and features complex questions and answers, primarily sourced from competitions and quizzes. MMLU (Hendrycks et al., 2021): This dataset comprises multiple-choice questions across various fields, assessing the model's knowledge mastery in academic and professional domains. OpenBookQA (Mihaylov et al., 2018): Focusing on scientific inquiries, this dataset requires reasoning rooted in principles and common sense. en-SQuAD-en2.0 (Rajpurkar et al., 2018): This dataset features question-answer pairs and evaluates reading comprehension skills. FreshQA-en (Vu et al., 2023): This dataset presents various question and answer types, offering a comprehensive assessment of QA capabilities. HotpotQA (Yang et al., 2018): This dataset consists of 113,000 Wikipedia based QA pairs that necessitate complex reasoning across multiple supporting documents and include sentence-level supporting facts.

Chinese Datasets. For the Chinese dataset, we utilize the following resources: WebQA (Chang et al., 2022): This open-domain QA dataset is collected via web crawlers, covering a wide array of topics and evaluating the performance of QA systems. SogouQA [5]: Provided by Sogou, this dataset features user-generated questions and system-generated answers, assessing accuracy and robustness. MLEC (Li et al., 2021): This dataset is designed to test the comprehension capabilities of models in various contexts. Xiezhi (Gu et al., 2024): A set of 249,587 Chinese/English questions covering 516 subjects for evaluating LLMs. SQuAD-zh [6]: The Chinese version of the English SQuAD (Rajpurkar et al., 2018) dataset serves to train and evaluate machine reading comprehension and QA systems. Chinese SimpleQA(CSQA) (He et al., 2024): A benchmark for evaluating the factuality of LLMs in short Chinese QA across diverse topics. CMMLU (Li et al., 2024a):This comprehensive benchmark assesses the knowledge and reasoning capabilities of language models across 67 topics, from basic to advanced.

[5]https://github.com/sherlcok314159/ChineseMRC-Data
[6]https://github.com/pluto-junzeng/ChineseSquad