Can Large Language Models Recall Reference Location Like Humans?

Anonymous ACL submission

Abstract

When completing knowledge-intensive tasks, humans sometimes need not just an answer but also a corresponding reference passage for auxiliary reading. Previous methods required obtaining pre-segmented article chunks through additional retrieval models. This paper explores leveraging the parameterized knowledge stored during the pre-training phase of large language models (LLMs) to independently recall reference passage from any starting position. We propose a two-stage framework that simulates the scenario of humans recalling easily forgotten references. Initially, the LLM is prompted to recall document title identifiers to obtain a coarse-grained document set. Then, based on the acquired coarse-grained document set, it recalls fine-grained passage. In the two-stage recall process, we use constrained decoding to ensure that content outside of the stored documents is not generated. To increase speed, we only recall a short prefix in the second stage, then locate its position to retrieve a complete passage. Experiments on KILT knowledge-sensitive tasks have verified that LLMs can independently recall reference passage location in various task forms, and the obtained reference significantly assist downstream tasks[1].

1 Introduction

Knowledge-intensive tasks, including open-domain question answering, dialogues, and fact verification, require access to a vast amount of world or domain-specific knowledge (Petroni et al., 2021). Common methods involve utilizing external knowledge sources such as Wikipedia and employing additional sparse or dense retrieval models to first retrieve relevant passage from Wikipedia as reference, then predict answers under the condition of the reference (Karpukhin et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021). However, traditional sparse lexical retrieval lacks deep semantic understanding, and in dual-tower dense retrieval models, the representations of questions and passages are usually obtained independently (Karpukhin et al., 2020), leading to only shallow interactions being captured (Khattab et al., 2021).

Recently, generative retrieval methods that leverage the generative capabilities of models (Cao et al., 2021; Tay et al., 2022; Bevilacqua et al., 2022; Wang et al., 2022; Lee et al., 2022) have gained increasing attention. They generate passage identifiers through autoregressive generative models, performing deep token-level cross-attention, and interacting with the entire parameter space of models trained on the target corpus to overcome bottlenecks. These retrieval methods have proven effective across various domains and tasks. However, both dense retrieval and previous generative retrieval methods can only obtain predefined and segmented passages (e.g., 100 words), making it difficult to freely choose the starting point of the reference. This somewhat affects the flexibility of obtaining references and the naturalness of reading. Humans, on the other hand, can recall any seen passage from memory and naturally recall it from any starting point.

Large language models (LLMs) have proven to possess strong generative understanding capabilities that can generalize to multiple tasks through prompting, and they have parameterized memory of extensive document knowledge during the pre-training phase. We explore whether LLMs can independently recall seen passage as reference from any starting position through prompting. To this end, we mimic the human process of recalling references: first recalling related coarse-grained documents or pages, then recalling fine-grained reference passage, as shown in Figure 1. Specifically, we employ a two-stage framework. In the first stage, we prompt the large language model (LLM) to recall document title identifiers, using a prefix

[1] Code is available at https://anonymous.4open.science/r/LLMRefLoc-3415.

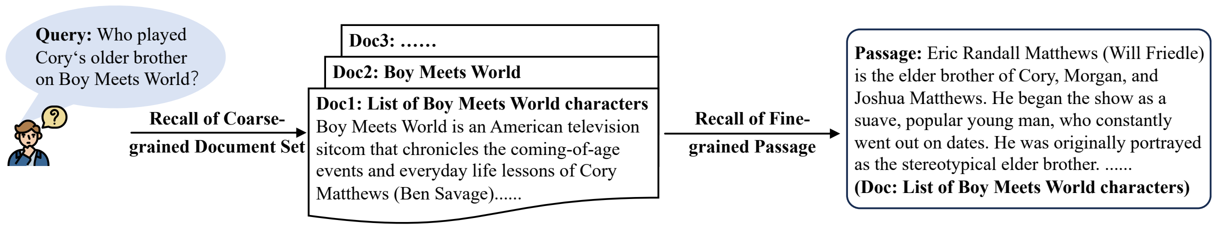

Query: Who played Cory's older brother on Boy Meets World?
Recall of Coarse-grained Document Set
Doc3: ......
Doc2: Boy Meets World
Doc1: List of Boy Meets World characters
Boy Meets World is an American television sitcom that chronicles the coming-of-age events and everyday life lessons of Cory Matthews (Ben Savage)......
Recall of Fine-grained Passage
Passage: Eric Randall Matthews (Will Friedle) is the elder brother of Cory, Morgan, and Joshua Matthews. He began the show as a suave, popular young man, who constantly went out on dates. He was originally portrayed as the stereotypical elder brother. ......
(Doc: List of Boy Meets World characters)

Figure 1: The common process of human recall of easily forgotten information often involves an intermediate step, first recalling easily memorable documents, then recalling to find the passage.

tree (Trie) (Cormen et al., 2022) to constrain decoding and ensure the recalled titles exist within the knowledge source. In the second stage, we construct a new FM-index (Ferragina and Manzini, 2000) with the high-ranking document set obtained from the first stage, which can effectively identify any substring within the set, ensuring that we can start constrained generation from any location in the document. We prompt the LLM to recall the required reference passage under this FM-index constraint. To leverage information from both stages of recall, we ultimately use a weighted score from the two-stage constrained decoding to select the final reference.

Although LLMs have impressive capabilities, generating complete reference passage may be too time-consuming. To address this issue, we explore whether LLMs can locate passage by recalling a prefix only. First, we prompt the LLM to generate a short prefix, then use the FM-index to locate the document containing this prefix within the document set. Finally, we use the Knuth-Morris-Pratt (KMP) algorithm to identify the starting position of the prefix in the document, determining a longer passage as the final reference from this position. We find that this method significantly speeds up the LLM's recall of reference. And the location of such a short prefix still maintains a high quality of reference retrieval. Our framework requires no additional retrieval models and pre-segmentation, allowing the LLM to independently and naturally obtain natural reference from articles of any length, ultimately naming it **LLMRefLoc**.

We conduct comprehensive experiments on 6 knowledge-sensitive tasks from the KILT benchmark (Petroni et al., 2021). We find that in a zero-shot setting, merely prompting the open-source LLMs Llama (Touvron et al., 2023a) and Llama-2 (Touvron et al., 2023b) enables the effective recall of needed page documents and fine-grained reference passage under the **LLMRefLoc** framework, enhancing the performance of downstream tasks. In summary, our main contributions are as follows:

- We explore for the first time the capability of LLMs to recall fine-grained reference passage under constraints with prompting.
- We propose a two-stage framework that mimics the human information search process, first recalling coarse-grained documents, then recalling fine-grained reference.
- By merely recalling a short prefix and locating it, we significantly speed up the recall of reference and verify the ability of LLMs to recall and locate.
- Across 6 knowledge-sensitive tasks, our framework excels in page and passage-level evaluations and can significantly improve the performance of downstream tasks.

2 LLMRefLoc

In this section, we detail our two-stage framework, **LLMRefLoc**. In the first stage, we prompt the LLM to recall title identifiers, which serve as candidate documents for the next stage. In the second stage, the LLM is prompted to recall reference passage from the documents obtained in the first stage. To increase speed, we only recall a short prefix, then locate and extract the reference within the document. The structure is shown in Figure 2.

2.1 First Stage: Coarse-Grained Document Recall

When facing knowledge-sensitive tasks, due to the difficulty of directly recalling fine-grained reference passage, as seen in the experimental analysis of directly recalling fine-grained passage in subsection 3.3, we can divide the recall process into two stages. Initially, we obtain a set of documents that are easier to recall, such as Wikipedia pages.

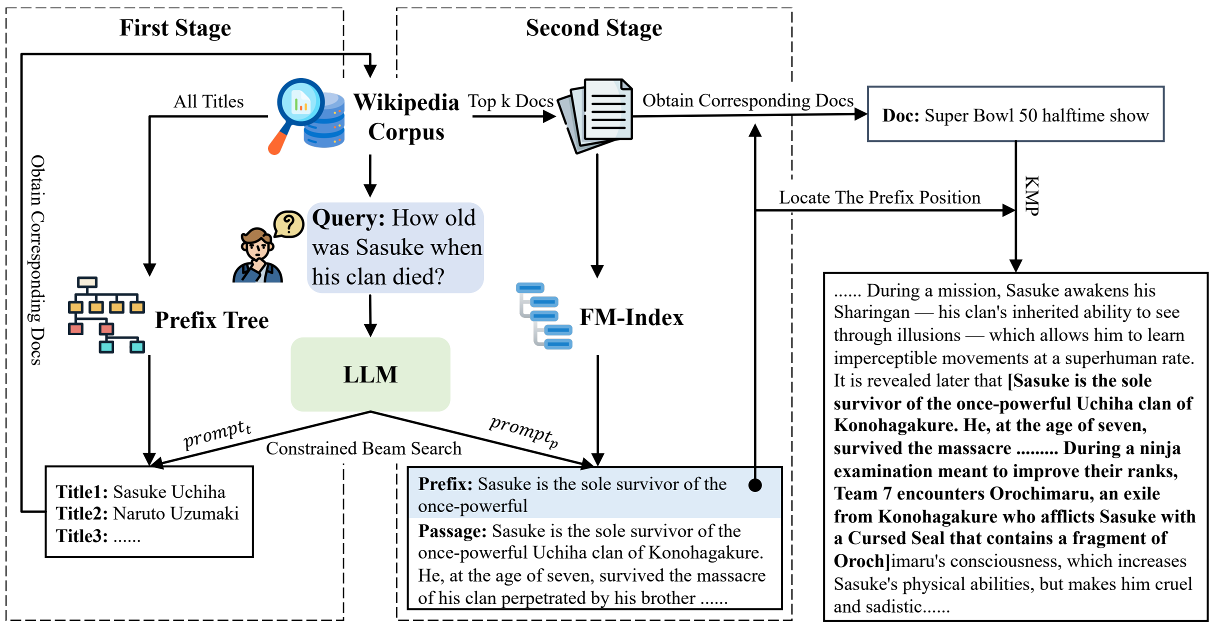

First Stage
Second Stage
All Titles
Wikipedia Corpus
Top k Docs
Obtain Corresponding Docs
Doc: Super Bowl 50 halftime show
Obtain Corresponding Docs
Locate The Prefix Position
KMP
Query: How old was Sasuke when his clan died?
Prefix Tree
FM-Index
LLM
$prompt_t$
$prompt_p$
Constrained Beam Search
Title1: Sasuke Uchiha
Title2: Naruto Uzumaki
Title3: ......
Prefix: Sasuke is the sole survivor of the once-powerful
Passage: Sasuke is the sole survivor of the once-powerful Uchiha clan of Konohagakure. He, at the age of seven, survived the massacre of his clan perpetrated by his brother ......
...... During a mission, Sasuke awakens his Sharingan — his clan's inherited ability to see through illusions — which allows him to learn imperceptible movements at a superhuman rate. It is revealed later that [Sasuke is the sole survivor of the once-powerful Uchiha clan of Konohagakure. He, at the age of seven, survived the massacre .... During a ninja examination meant to improve their ranks, Team 7 encounters Orochimaru, an exile from Konohagakure who afflicts Sasuke with a Cursed Seal that contains a fragment of Oroch]imaru's consciousness, which increases Sasuke's physical abilities, but makes him cruel and sadistic......

Figure 2: Shows the **LLMRefLoc** framework. First, all Wikipedia titles are stored in a prefix tree, then the LLM is prompted to recall title identifiers under this prefix tree constraint. Subsequently, an FM-index is constructed from the top k documents obtained, and the LLM recalls reference passage under the new constraint. The grey area represents the short prefix, used for the next step of locating and extracting the complete passage.

For Wikipedia page documents with titles, we can directly prompt the LLM to recall existing title identifiers, uniquely determining the title through constrained generation. This method leverages the knowledge stored within the LLM and can also ensure that the generated title is uniquely existing.

For the input query x, we use the prompt $prompt_t(x)$ to stimulate the LLM to recall Wikipedia titles that could potentially cover the content of the query. For example, for open-domain question-answering tasks, we can simply design the prompt as: "Question: \n\nThe Wikipedia title corresponding to the above question is:\n\nTitle:".

We define the set of all Wikipedia titles as T and the set of all documents as D, such that each title and document uniquely correspond to each other. First, we store all Wikipedia titles T in a prefix tree (Trie) (Cormen et al., 2022). At each step in the autoregressive generation of the LLM, we determine the set of tokens to be generated next by using both the prefix tree and the previously generated tokens, masking the logits of tokens not belonging to this set as $-\infty$. In other words, the prefix tree acts as a navigation structure, guiding the model to generate tokens along a path corresponding to a known title t in the set T. A detailed introduction to Trie can be found in Appendix A.1. We compute the score of autoregressive generation through the default implementation in the (Wolf et al., 2020) library, with the score of title t given $prompt_t(x)$ as:

$$\begin{aligned} &score_1(t|prompt_t(x)) \\ &= \frac{\log p_\theta(y_t|prompt_t(x))}{|y_t|} \\ &= \frac{\sum_{i=1}^{l_t} \log p_\theta(y_i|y_{<i}, prompt_t(x))}{l_t} \end{aligned} \tag{1}$$

where y_t represents the set of tokens in the title t, l_t and $|y_t|$ represent the number of tokens generating the title identifier, θ is the model's parameters.

2.2 Second Stage: Fine-Grained Passage Recall

To enable LLMs to freely recall reference passage from any starting position, we utilize the FM-index constraint (Ferragina and Manzini, 2000). The FM-index can be considered a specialized prefix tree that supports searching from any position. Given a starting token or string, the FM-index can provide a possible list of token successors within $O(Vlog(V))$ time, where V is the size of the vocabulary. A detailed introduction to the FM-index can be found in Appendix A.2. Here, the FM-index can be pre-built for all documents, ready to be retrieved when needed to omit the time for additional construction.

Specifically, after obtaining the coarse-grained document set, we construct a new FM-index for the document set D_k corresponding to the top k title identifiers obtained. The new document set constraint, compared to recalling from the entire document set, significantly reduces the possible generation space, making the recall of fine-grained passage more manageable. At this stage, we prompt the LLM with $prompt_p$ to generate the passage p under constraint, for example, for open-domain question-answering we can simply design the prompt as “Question: \n\nThe answer to the above question can be found in the following Wikipedia paragraph:\n\nAnswer:" to have the LLM recall the corresponding reference passage. We determine the allowed set of tokens for subsequent generation based on the previously generated part, enabling us to start generating the complete passage p from any position in the document set D_k. We measure the score of the task corresponding passage by using the autoregressive formula to calculate the score:

$$\begin{aligned} & score_2(p|prompt_p(x)) \\ & = \frac{\log p_\theta(y_p|prompt_p(x))}{|y_p|} \\ & = \frac{\sum_{i=1}^{l_p} \log p_\theta(y_i|y_{<i}, prompt_p(x))}{l_p} \end{aligned} \tag{2}$$

where y_p represents the set of tokens in the passage p, θ is the model parameters, $|y_p|$ and l_p represent the number of tokens generating the passage, usually set between 150 to 200. To integrate information generated from both stages, we calculate the weighted sum of the scores from the first and second stages to obtain the final score under input query x:

$$\begin{aligned} score(p|x) = \alpha * score_1(t|prompt_t(x)) \\ + (1-\alpha) * score_2(p|prompt_p(x)) \end{aligned} \tag{3}$$

where $score_1(t|prompt_t(x))$ is the score of the Wikipedia page title t corresponding to the passage p, α is a hyperparameter controlling the weight of the two stages. Finally, among all recalled passages, the one with the highest $score(p|x)$ value is selected as the best reference.

2.3 Short Prefix Recall and Localization

Although LLMs can effectively recall longer reference passage directly, their expensive inference speed somewhat diminishes their application value. In fact, generating a passage of about 150 to 200 tokens requires a significant amount of computational resources and time, which is unacceptable in many real-world scenarios. The length of generation is a key factor hindering its speed. To address this challenge, we explore whether LLMs can locate reference passage by merely recalling a prefix. For this, we propose the method Short Prefix Recall Location (SPRL). The basic idea of SPRL is to first prompt the generation of a relatively short segment prefix, then locate and extract the complete passage containing this prefix within the source document set. Specifically, we divide the entire process into the following two main steps.

Initially, given an input question q, we prompt the LLM to recall a short text prefix p_s, consisting of l_{p_s} tokens, guided by the prompt $prompt_p$, which is the same as used in the second stage. In this step, we set l_{p_s} significantly smaller than the full length of the long passage, thereby saving significant generation time and computational resources.

Subsequently, we find the document d corresponding to p_s within the document set D_k obtained in the first stage. Since we have controlled the number of documents in the document set D_k obtained in the first stage, in most cases, we can obtain a unique document d containing p_s. When we cannot obtain a unique document, we simply default to selecting the first document. In the next step, we use the Knuth-Morris-Pratt (KMP) string matching algorithm to quickly determine the starting position st of p_s in d, then extract a complete reference $p_{final} = d[st : st + l_p]$ starting from st, with a length of l_p tokens. The autoregressive score of the short prefix is also calculated using Equation 2 and the final two-stage score is obtained using Equation 3 to select the final reference. In Experiment 3.2.1, we find that recalling just the prefix for localization still achieves good results in obtaining reference.

3 Experiments

In this section, we conduct comprehensive experiments on coarse-grained pages, fine-grained passage-level reference evaluation, and downstream tasks to validate the effectiveness of our framework. Additionally, we perform further analyses and experiments.

3.1 Experimental Setup

Datasets We conduct extensive experiments on 6 knowledge-sensitive tasks from the KILT benchmark (Petroni et al., 2021). These tasks include open-domain question answering tasks such as NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and ELI5 (Fan et al., 2019), the fact-checking task FEVER (Thorne et al., 2018), and the open-domain dialogue system WoW (Dinan et al., 2018). Due to the lack of publicly available test sets for the KILT dataset, all experiments are conducted on its validation set. For specific details about the datasets, please refer to Appendix B. We evaluate the quality of coarse-grained pages and fine-grained reference passage, as well as the enhancement of reference for downstream tasks.

Evaluation Metrics We use R-Precision as the evaluation metric for coarse-grained page-level. For fine-grained reference evaluation, since we generate passage from any position in the document rather than retrieving reference from pre-segmented passages, to standardize evaluation criteria, for datasets such as NQ, TriviaQA, and HotpotQA, we calculate the percentage of references containing at least one gold standard answer, termed Answer in Context. In other datasets, we measure the percentage of references containing at least one gold standard entity, termed Entity in Context. For downstream tasks, we use the metrics officially implemented by KILT: Exact Match (EM) for NQ, TriviaQA, and HotpotQA; Rouge-L for ELI5; accuracy for FEVER; and F1 for WoW.

Baseline Models We compare with several traditional retrieval models. These models all use the passage segmentation from the official KILT as the source for obtaining reference. For unsupervised retrieval models, we compare the traditional sparse retrieval model BM25[2] (Robertson et al., 2009), and the dense retrieval model Contriever (Izacard et al., 2022). We also compare with the dense retrieval model DPR[3] (Karpukhin et al., 2020) that has been fine-tuned on the full dataset. We input the first passage retrieved by the model as the reference context into the LLM, which then reads the relevant reference to answer downstream tasks.

Implementation Details We choose the 7b and 13b versions of the open-source LLMs Llama (Touvron et al., 2023a) and Llama-2 (Touvron et al., 2023b) for reference recall. Downstream tasks use Llama-2-13b as the reading model. We merge the passage fragments from KILT into complete documents, serving as the data source for recall. The length of the complete documents is arbitrary. In the recall phase, we always use a beam search generation strategy. In the first stage of generation, the beam size is set to 15, and we construct an FM-index containing the top $k = 2$ documents. In the second stage, the beam size is set to 10, the length of the short prefix is $l_{p_s} = 16$, and we extract a token length of $l_p = 150$ as the final reference. The weight setting for the two-stage weighted method is $\alpha = 0.9$. All downstream tasks use greedy decoding. The prompts used in the experiments can be found in Appendix C. All experimental results are from a single run on Tesla A100 40G GPUs.

3.2 Experimental Results

3.2.1 Page-level Results

Coarse-grained page-level results, as shown in Table 1, demonstrate that the LLMRefLoc framework, when implemented with Llama-2-13b, achieves the best R-precision scores of 57.77, 48.70, 83.69, and 57.63 on the NQ, HotpotQA, FEVER, and WoW datasets, respectively. This significantly surpasses the performance of sparse retrieval BM25 and dense retrieval Contriever in a zero-shot scenario. It also shows strong competitive power against the fully trained DPR method, especially on the WoW and FEVER datasets, with improvements of 27.08 and 31.01 points, respectively. This result is consistent with the hypothesis that LLMs are powerful in recalling coarse-grained title identifiers, enabling the acquisition of high-quality relevant pages that assist in the subsequent fine-grained recall stage.

3.2.2 Passage-level Results

Fine-grained reference passage results, as shown in Table 2, reveal that the LLMRefLoc framework, when implemented with Llama-2-13b, also achieves the best scores of 68.20, 30.04, 58.42, and 63.43 on the TriviaQA, HotpotQA, FEVER, and WoW datasets, respectively. We note that the improvement of the framework in fine-grained reference passage compared to the DPR method is relatively reduced compared to the page-level results. This suggests potential for optimization in activating LLMs to recall more detailed and longer reference, presenting a greater challenge compared to recalling shorter title. Notably, DPR performs excellently on the NQ dataset, which may relate to its training data format. Interestingly, in the

[2] We implement BM25 retrieval using the https://github.com/castorini/pyserini repository

[3] We conduct experiments with the trained DPR model and preprocessed vector index from the https://github.com/facebookresearch/KILT repository.

Method	Open-domain QA				Fact Check.	Dial.
	NQ	TriviaQA	HotpotQA	ELI5	FEVER	WoW
Contriever	34.72	34.28	26.14	11.02	55.64	29.67
BM25	26.33	31.78	41.30	6.83	52.09	28.78
DPR*	54.74	45.68	25.46	**16.19**	56.61	26.62
LLMRefLoc(Llama-7b)	54.46	**57.03**	44.56	15.13	76.57	52.91
LLMRefLoc(Llama-13b)	54.42	55.53	46.30	12.94	77.55	34.51
LLMRefLoc(Llama-2-7b)	56.33	56.43	46.20	14.60	77.29	49.61
LLMRefLoc(Llama-2-13b)	**57.77**	54.41	**48.70**	15.00	**83.69**	**57.63**

Table 1: Coarse-grained page-level results, measured by R-Precision. $\star$ indicates full data training. Bold data in the table represents the best results, while underlined data indicates the second-best results.

Method	Open-domain QA				Fact Check.	Dial.
	NQ	TriviaQA	HotpotQA	ELI5	FEVER	WoW
	Answer in Context			Entity in Context		
Contriever	19.28	37.21	11.16	12.48	40.48	45.15
BM25	23.65	58.87	29.45	12.01	58.33	50.36
DPR*	**47.94**	66.60	20.29	14.40	41.22	45.38
LLMRefLoc(Llama-7b)	36.87	58.48	25.55	15.99	54.85	59.40
LLMRefLoc(Llama-13b)	37.72	60.96	26.34	14.80	55.20	50.79
LLMRefLoc(Llama-2-7b)	38.07	62.88	27.55	**16.85**	56.23	57.79
LLMRefLoc(Llama-2-13b)	40.82	**68.20**	**30.04**	15.06	**58.42**	**63.43**

Table 2: Fine-grained passage-level results, measured by Answer in Context and Entity in Context of the top 1 reference. $\star$ indicates full data training. Bold data in the table represents the best results, while underlined data indicates the second-best results.

HotpotQA dataset, BM25 remains competitive, surpassing dense retrieval methods, possibly due to the longer questions in this dataset leading to more vocabulary overlap. LLMRefLoc shows significant progress on the FEVER and WoW datasets, demonstrating the potential and adaptability of LLMs in recalling high-quality reference passage across different task formats. Furthermore, the general enhancement in performance with the progression from Llama to Llama-2 and the increase in model size indicates a correlation between the recall ability and the underlying capabilities of LLMs.

3.2.3 Downstream Task Results

Downstream task results are presented in Table 3. LLMRefLoc, based on Llama-2-13b recalled passage, achieved the best scores of 72.94, 78.79, and 14.77 on the TriviaQA, FEVER, and WoW downstream tasks, respectively, validating the performance of LLM recall references in downstream tasks. On the open-domain question answering NQ dataset, although DPR performed excellently after full data training, LLMRefLoc also displayed highly competitive performance. On the other hand, in the TriviaQA and HotpotQA datasets, due to the length of the questions, BM25 achieved excellent performance by obtaining more vocabulary overlap, yet LLMRefLoc still achieved comparable or better performance in most cases. The unsupervised trained Contriever performed relatively poorly across all tasks, emphasizing the crucial role of supervised training in enhancing the performance of dense retrieval models.

3.3 Ablation Study

In this subsection, we conduct ablation studies, comparing methods without weighted scores (w/o weight), without Short Prefix Recall Location (w/o SPRL), and without the first stage of document title recall (w/o first stage). The results are shown in Table 4.

Without weighted scores, relying solely on the recall scores from the second stage leads to a simul-

Method	Open-domain QA				Fact Check.	Dial.
	NQ	TriviaQA	HotpotQA	ELI5	FEVER	WoW
	EM			R-L	ACC	F1
Llama-2-13b	19.74	68.71	15.64	19.46	73.23	13.90
Contriever	24.78	69.25	20.34	<u>20.71</u>	73.61	13.96
BM25	25.84	71.49	**27.23**	20.48	77.54	14.02
DPR*	**33.49**	<u>72.68</u>	23.13	**20.75**	75.27	14.17
LLMRefLoc(Llama-7b)	29.78	70.18	24.61	20.60	78.10	14.47
LLMRefLoc(Llama-13b)	29.68	71.60	25.48	20.24	<u>78.53</u>	14.33
LLMRefLoc(Llama-2-7b)	29.89	70.04	25.55	20.50	78.04	<u>14.48</u>
LLMRefLoc(Llama-2-13b)	<u>31.69</u>	**72.94**	<u>26.13</u>	20.61	**78.79**	**14.77**

Table 3: Downstream task results. $\star$ indicates that the retrieval model underwent full data training. Bold data in the table represents the best results, while underlined data indicates the second-best results.

Method	NQ	TriviaQA	HotpotQA	NQ	TriviaQA	HotpotQA
	R-Precision			Answer in Context		
LLMRefLoc	57.77	54.41	48.70	40.82	68.20	30.04
w/o weight	51.22	49.23	48.70	39.06	66.86	28.88
w/o SPRL	55.30	51.50	48.70	37.43	64.64	26.18
w/o first stage	32.22	24.87	23.36	36.27	63.33	24.16

Table 4: Ablation study results on the NQ, TriviaQA, and HotpotQA datasets are shown, with the left half showing the R-Precision at the coarse-grained page level and the right half showing Answer in Context for fine-grained passage. We compared the performance differences without weighted scores, without SPRL, and without the first stage.

taneous decrease in performance for both coarse and fine-grained results, emphasizing the importance of considering scores from both stages. The model, by taking into account title scores, is more capable of selecting the correct document, and within the correct document, it is more likely to choose the correct reference. More results on the choice of weighted α can be found in Appendix D.

Without SPRL, recalling longer segments has a minor impact on page-level performance. However, it significantly affects the quality of fine-grained reference passage, where longer recall lengths paradoxically lead to decreased performance. This result is somewhat counterintuitive and might be due to all document knowledge being stored in the parameters during the pre-training phase, with a short prefix sufficient to locate the required reference. Longer references introduce redundancy and noise, thus lowering effectiveness. Notably, when using Llama-2-13b for recall, recalling complete passages on the NQ dataset takes about 600 minutes, while recalling short prefixes only requires 150 minutes, significantly reducing time costs. However, considering that dense retrieval takes about 20 minutes, further optimization of speed remains crucial. More experiments on prefix length can be found in Appendix E.

Without the first stage of document title recall, the quality of reference further declines, significantly impacting the quality of page retrieval. This indicates that using LLMs to directly recall references across a vast number of documents has considerable limitations and opportunities for improvement. The ability of merely prompting LLMs to recall and locate fine-grained reference passage is very limited, making the first stage of recalling document title identifiers crucial.

3.4 Further Analysis

After General Fine-tuning of LLMs We also test the Vicuna model (Chiang et al., 2023) and the Llama-2-chat model refined through reinforcement learning from human feedback (Touvron et al., 2023b), both of which underwent general fine-tuning. This general fine-tuning did not significantly enhance the performance of LLMs in recalling and locating reference. This may be due to the paradigm difference between the fine-tuning data and the recall location task, coupled with the fact that most knowledge was already acquired during the pre-training phase. By creating more diverse recall instruction tuning data, further improvements in model performance might be achieved. Detailed results can be found in Appendix F.

Impact of Few-Shot We explore adding few-shot prompts in the second stage of fine-grained recall and observed its impact on overall performance. This approach brought slight improvements only in the HotpotQA dataset, while showing a slight decline in NQ and TriviaQA. Importantly, adding more few-shot examples significantly reduced generation speed. This suggests that, although few-shot prompting offers a potential path for improvement, extensive exploration is still needed to devise more effective prompting methods. Detailed results can be found in Appendix G.

Memory Usage Analysis Dense retrieval methods such as Contriever and DPR require over 60GB of memory usage. In contrast, sparse retrieval methods use far less memory, only needing 17GB. The LLMRefLoc framework, utilizing FM-index and Trie indexing, requires only 8GB when pre-encoding and storing all documents with FM-index, and the Trie storing all title identifiers needs just 25MB, which is negligible. Compared to sparse and dense retrieval methods, the recall framework effectively saves memory.

4 Related Work

Traditional methods of obtaining reference include sparse and dense retrieval. Sparse retrieval, using TF-IDF and BM25, matches questions and passages (Robertson et al., 2009; Chen et al., 2017; Yang et al., 2019). Recent approaches, such as ORQA (Lee et al., 2019) and DPR (Karpukhin et al., 2020), employ dense context vectors for passage indexing to enhance performance. However, in dual-encoder dense retrieval models, the representations of questions and passages are obtained independently, leading to performance limitations due to shallow vector interactions (Khattab and Zaharia, 2020).

Interest has surged in using autoregressive language models to generate identifiers to simplify the retrieval process and address the bottleneck of limited interactions in dual-encoder models. For example, Cao et al. (2021) generates Wikipedia page titles for retrieval, Tay et al. (2022) targets generating paths from root to leaf in hierarchical clustering trees, and Bevilacqua et al. (2022) maps to unique n-grams. Recently, Lee et al. (2022) proposed a generative multi-hop retrieval method, Li et al. (2023b) employed multiple identifiers to collaboratively determine retrieval passages, and Ren et al. (2023) introduced a two-stage approach that first generates passages, then URL identifiers. Ziems et al. (2023) utilized LLMs to first generate page URLs, followed by traditional retrieval techniques to obtain fine-grained passages. However, whether traditional or generative retrieval, they can only obtain predefined and segmented text chunks, making it difficult to naturally retrieve references from arbitrary positions.

Recent research has found that relevant knowledge can be extracted from LLMs through prompting, especially in domains with insufficient coverage in knowledge bases (Liu et al., 2022; Fang et al., 2022). Enhancing model performance through the output of LLMs has also gained attention. Liu et al. (2022); Sun et al. (2023); Yu et al. (2023) propose using GPT-3 to generate relevant context as references, treating these contexts as additional inputs when answering questions. However, fully generating context through LLMs is still plagued by the phenomenon of hallucination (Li et al., 2023a).

5 Conclusion and Future Work

This paper introduces a framework named **LLMRefLoc**, which utilizes LLMs to independently recall reference passage and can be flexibly applied to various knowledge-sensitive tasks. Mimicking the human habit of searching for information, we first prompt the LLM to recall relevant document pages, and then from these pages, recall and locate reference passage. Furthermore, through beam search constrained by Trie and FM-index, we ensure that the content recalled by the LLM is a subset of existing texts. This framework can be flexibly used with various open-source LLMs, expanding the potential applications of LLMs. In the future, we consider enhancing the ability of LLMs to recall under constraints through instruction tuning, as well as applying this framework to more retrieval fields, exploring lightweight methods to inject new document knowledge into LLMs, and integrating multi-hop reasoning into reference recall.

Limitations

Although LLMRefLoc demonstrates the potential of LLMs to recall reference passage in knowledge-sensitive tasks like humans, its application still faces several limitations. Firstly, this framework struggles to surpass the performance of the current state-of-the-art retrieval models, especially those models that have been fine-tuned on specific tasks through supervision. In the future, there is a need to explore more effective ways of instruction tuning for recalling under constraints. At the same time, LLMRefLoc relies on document title identifiers for phased recall, meaning that its recall capability may be limited for documents lacking clear titles or identifiers.

Moreover, the framework finds it challenging to effectively recall documents that appear less frequently in the pre-training stage. This indicates that if a document appears infrequently in the training data of the LLM, or if the document content significantly differs from the training data, LLMRefLoc may encounter difficulties in recalling these documents. For the updating of documents and the injection of new knowledge, LLMRefLoc requires additional training to inject this new information into the model parameters. There is still a need to explore more efficient, lightweight methods for injecting new documents in the future.

Ethical Considerations

Our framework ensures that the generated content is entirely derived from reference materials, with Wikipedia as an example in this paper, thus not introducing additional significant ethical issues. However, in practical applications, we must ensure that the source document set relied upon is harmless to prevent the spread of inaccurate or harmful information.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Michael Burrows. 1994. A block-sorting lossless data compression algorithm. *SRS Research Report*, 124.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms*. MIT press.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Yuwei Fang, Shuohang Wang, Yichong Xu, Ruochen Xu, Siqi Sun, Chenguang Zhu, and Michael Zeng. 2022. Leveraging knowledge in multilingual commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3237–3246.

Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *Proceedings 41st annual symposium on foundations of computer science*, pages 390–398. IEEE.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and

Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, 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.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022. Generative multi-hop retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1436, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv e-prints*, pages arXiv–2305.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023b. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6636–6648, Toronto, Canada. Association for Computational Linguistics.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169.

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

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6102–6114, Toronto, Canada. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations*.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.

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

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77.

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.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Noah Ziems, Wenhao Yu, Zhihan Zhang, and Meng Jiang. 2023. Large language models are built-in autoregressive search engines. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2666–2678, Toronto, Canada. Association for Computational Linguistics.

A Constrained Decoding Methods

A.1 Trie

The Trie (Cormen et al., 2022), also known as a dictionary tree or prefix tree, is a tree-like data structure used to store an associative array where the keys are usually strings. Unlike a binary search tree, keys in a Trie are not stored directly within the nodes; instead, they are determined by the node's position in the tree. All descendants of a node have the same prefix, associated with the string corresponding to that node.

The overall process during constrained decoding using a Trie is shown in Figure 3a. Taking the generation of the title "Testamentary Capacity" as an example, the LLM first selects "Testament" from the set of token strings that start all titles. Subsequently, we can obtain the set of token strings {and, ary} following the string "Testament". After the LLM selects "ary", we get the prefix string "Testamentary", and finally continue to select new strings from the next set of token strings until the end-of-sequence token </s>is encountered, ceasing generation.

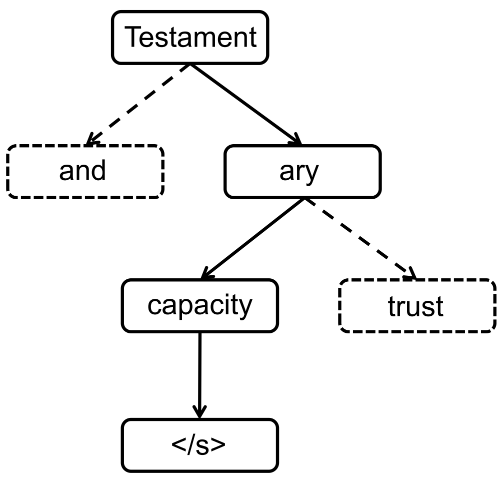
A.2 FM-Index

The FM-index (Ferragina and Manzini, 2000) is a data structure used for text retrieval that can store text efficiently with linear space complexity and support fast substring search operations. It is constructed based on the Burrows-Wheeler Transform (BWT) (Burrows, 1994). BWT is a method that converts a string into a form that is easy to compress. Given a string, BWT produces a transformed string through the following steps: generate all cyclic shifts of the string, sort all these shifts lexicographically, take the last character of each sorted shifted string to form a new string, which is the BWT result. For example, for the string "CABAC", the process of building the FM-index is as follows:

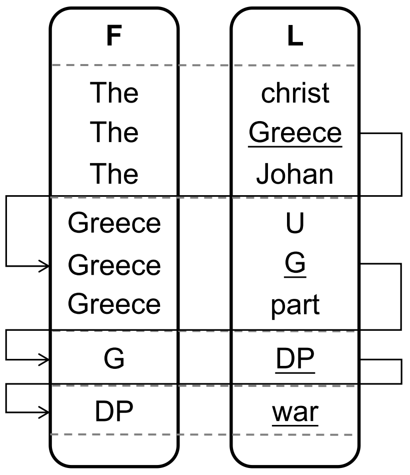
$$\begin{array}{cccccc}
\textbf{F} & & & & & \textbf{L} \\
\$^6 & C & A & B & A & C^5 \\
A^2 & B & A & C & \$ & C^1 \\
A^4 & C & \$ & C & A & B^3 \\
B^3 & A & C & \$ & C & A^2 \\
C^5 & \$ & C & A & B & A^4 \\
C^1 & A & B & A & C & \$^6
\end{array}$$

where $ is a special string termination token, the numbers in the upper right corner of the letters in the F and L columns are the corresponding position index numbers. The FM-index explicitly stores two main parts: the F column and the L column. The F column is the lexicographically sorted characters of the transformed string, and the L column is the result of BWT. In addition, it stores additional position information to recover the original string from the BWT result. When we want to query a substring, the FM-index starts from the last character of the substring, using the information in the F column and the L column to gradually narrow down the possible position range until the exact position of the substring is determined or the substring is determined to be non-existent.

The overall process during constrained decoding using FM-index is shown in Figure 3b. Considering the generated prefix "The Greece GDP warrants are



(a)



(b)

Figure 3: (a) Shows the process of a LLM generating title identifiers using a prefix tree. (b) Shows the process of a LLM generating passage prefixes in a document set via FM-index.

Dataset	Task	Input Format	Output Format	Size
NQ	Open Domain QA	Question	Extractive	2837
HotpotQA	Open Domain QA	Question	Short Abstractive	5600
TriviaQA	Open Domain QA	Question	Extractive	5359
ELI5	Open Domain QA	Question	Long Abstractive	1507
FEVER	Fact Checking	Claim	Classification	10444
WoW	Dialogue	Conversation	Long Abstractive	3054

Table 5: Additional details of the datasets.

not technically bonds as investors do" for example, it first starts from the string "The" generated from all corpus, and gets its corresponding L column string set {christ, Greece, Johan}. After "Greece" is selected by the LLM, we can get the next set {U, G, part}, and continue the iteration until reaching the set maximum prefix length to stop generating.

B Dataset Details

- Natural Questions (NQ) (Kwiatkowski et al., 2019) is constructed from real anonymized aggregated queries submitted to the Google search engine, with answers being snippets from manually annotated Wikipedia articles.

- TriviaQA (Joshi et al., 2017) comprises a set of questions and answers initially crawled from knowledge question-answering and quiz websites.

- HotpotQA (Yang et al., 2018) contains a set of question-answer pairs based on Wikipedia, requiring multiple-step reasoning over multiple Wikipedia pages to answer each question.

- ELI5 (Fan et al., 2019) is a large-scale corpus for long-form question answering, consisting of questions and answers from the Reddit forum "Explain Like I'm Five" (ELI5), which require detailed and in-depth responses to open-ended questions.

- FEVER (Thorne et al., 2018) is one of the largest datasets for fact-checking, used to determine whether a statement is supported or refuted based on textual sources.

- Wizard of Wikipedia (WoW) (Dinan et al., 2018) is a task that requires intelligent agents in open-domain dialogues to demonstrate knowledge usage. The dialogues are directly grounded in knowledge retrieved from Wikipedia.

All experiments are tested using the public validation set as divided in the official KILT. Addi-

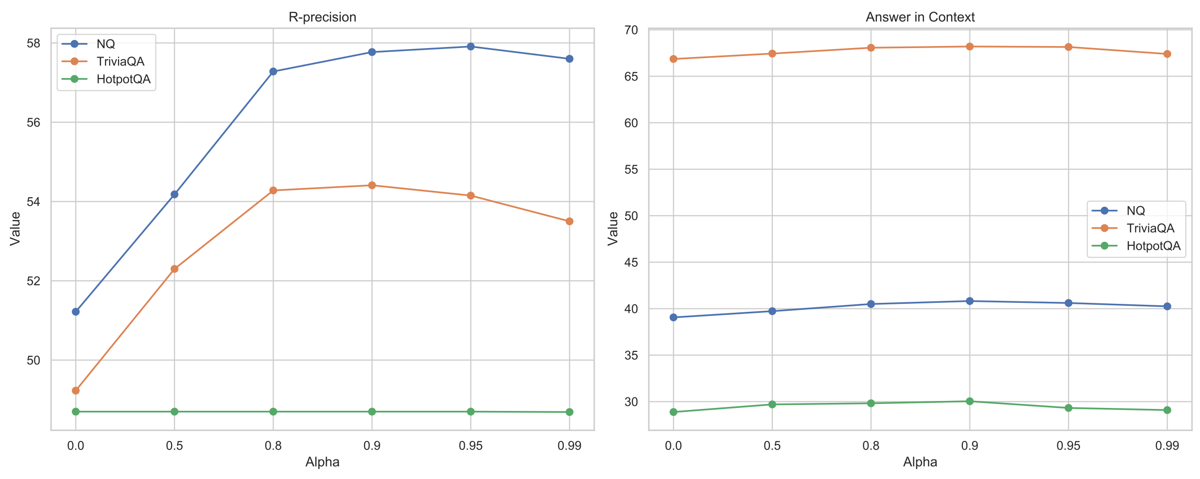


Figure 4: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for Llama-2-13b when setting α to {0.0,0.5,0.8,0.9,0.95,0.99}.

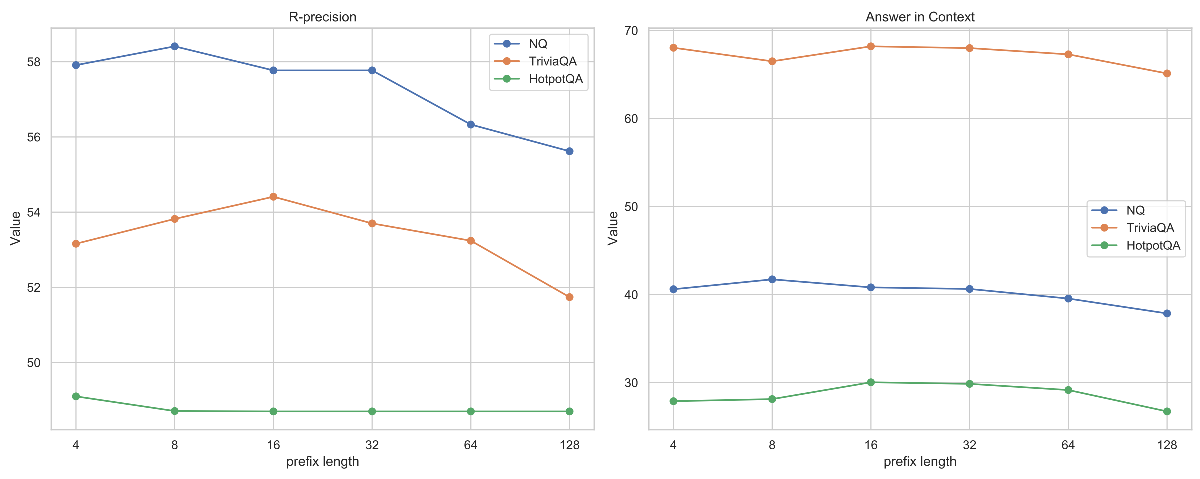


Figure 5: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for Llama-2-13b with different prefix token lengths l_{p_s} set to {4,8,16,32,64,128}.

tional details of the datasets are presented in Table 5. All the data used in this paper come from the KILT benchmark(Petroni et al., 2021), and KILT is MIT licensed[4].

C Prompts

In this subsection, we introduce the prompts used in the first stage for recalling coarse-grained title identifiers, in the second stage for recalling fine-grained reference passage, and in downstream tasks.

C.1 Prompts for the First Stage

- Open-domain Question Answering: "Question: { }\n\nThe Wikipedia article corresponding to the above question is:\n\nTitle:"

[4]https://opensource.org/licenses/MIT

- Fact Verification: "Claim: { }\n\nThe Wikipedia article corresponding to the above claim is:\n\nTitle:"

- Open-domain Dialogue System: "Conversation: { }\n\nThe Wikipedia article corresponding to the above conversation is:\n\nTitle:"

C.2 Prompts for the Second Stage

- Open-domain Question Answering: "Question: { }\n\nThe Wikipedia paragraph to answer the above question is:\n\nAnswer:"

- Fact Verification: "Claim: { }\n\nThe Wikipedia paragraph to support or refute the above claim is:\n\nAnswer:"

- Open-domain Dialogue System: "Conversa-

Method	NQ	TriviaQA	HotpotQA	NQ	TriviaQA	HotpotQA
	R-Precision			Answer in Context		
LLMRefLoc(Llama-7b)	52.56	55.35	43.88	34.72	55.96	24.43
LLMRefLoc(Vicuna-1.3-7b)	48.47	47.99	40.79	35.28	56.41	23.75
LLMRefLoc(Llama-13b)	51.53	56.62	46.09	36.55	61.28	26.43
LLMRefLoc(Vicuna-1.3-13b)	52.73	46.61	43.41	36.55	67.29	26.63
LLMRefLoc(Llama-2-7b)	56.26	56.52	46.20	38.03	62.87	27.48
LLMRefLoc(Llama-2-chat-7b)	3.31	1.12	0.98	4.09	3.97	3.43
LLMRefLoc(Vicuna-1.5-7b)	50.76	51.73	41.23	34.16	55.98	24.43
LLMRefLoc(Llama-2-13b)	57.77	54.41	48.70	40.82	68.20	30.04
LLMRefLoc(Llama-2-chat-13b)	1.94	1.60	1.55	6.38	7.93	4.71
LLMRefLoc(Vicuna-1.5-13b)	52.24	56.34	45.90	37.22	63.24	27.14

Table 6: On the NQ, TriviaQA, and HotpotQA datasets, experimental results after general fine-tuning of the model are presented. The left side shows the page-level R-Precision, while the right side displays the passage-level Answer in Context.

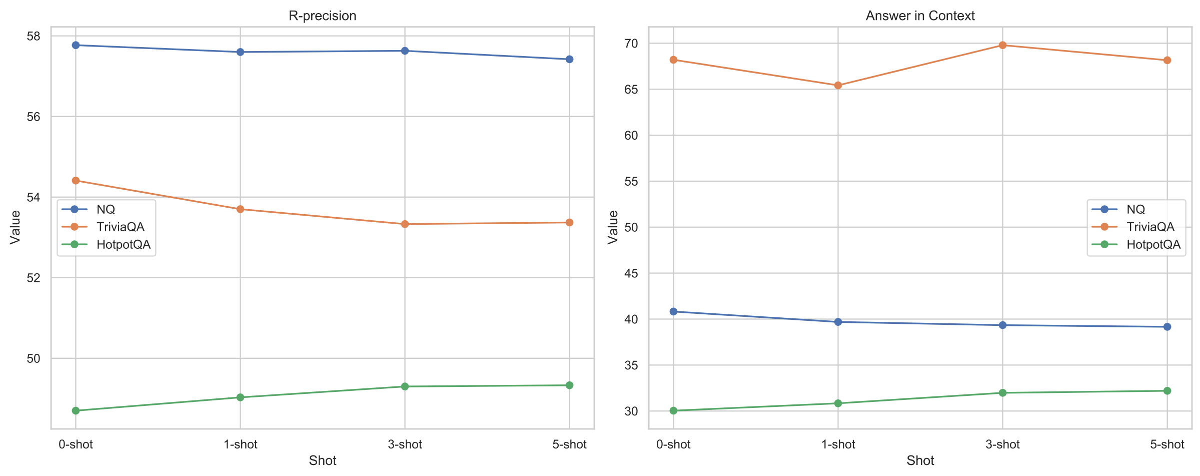


Figure 6: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for Llama-2-13b under {0,1,3,5}-shot few-shot prompts.

tion: {}\n\nThe Wikipedia paragraph to answer the above conversation is:\n\nAnswer:"

C.3 Prompts for Reading Comprehension

- Open-domain Question Answering (NQ, TriviaQA, HotpotQA): “Refer to the passage below and answer the following question with just a few words.\nPassage: {}\nQ: {}\nA: The answer is"

- Open-domain Question Answering (ELI5): “Refer to the passage below and answer the following question in detail.\nPassage: {}\nQ: {}\nA:"

- Fact Verification: “background: {}\nclaim: {}\nQ: Is the claim true or false?\nA:"

- Open-domain Dialogue System: “background: {}\n{}\n"

D Experimental Results for Different Values of Alpha

In Figure 4, we compare the experimental results of LLMRefLoc when implemented based on Llama-2-13b with different α values. When $\alpha = 0.0$, it’s equivalent to not having a two-stage weighted method, relying only on the scores from the second stage’s fine-grained passage recall, resulting in the selection of suboptimal reference. With the increase of α, the model sees improvements in both page-level and passage-level results, proving the importance of the first stage document scores for

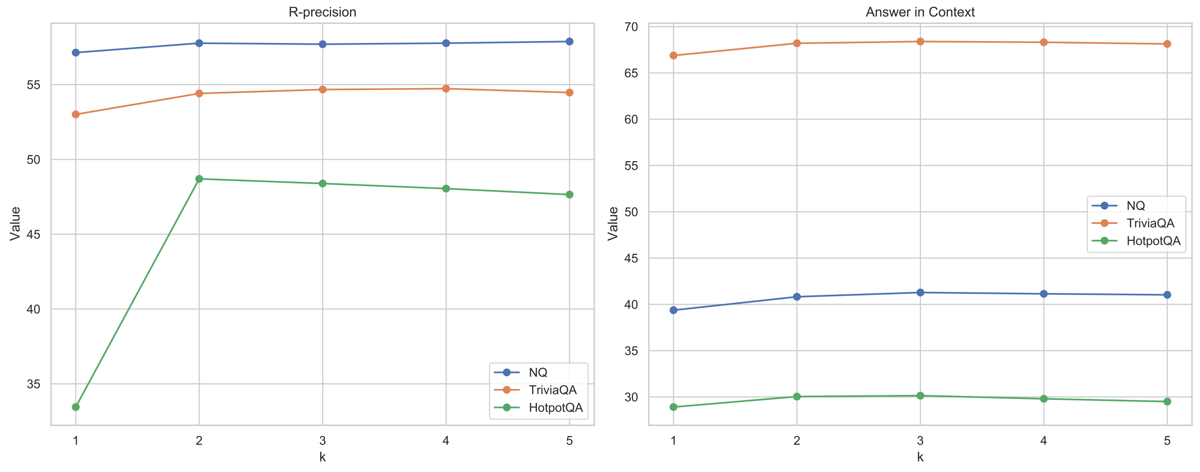


Figure 7: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for Llama-2-13b with the number of documents selected in the first stage k set to {1,2,3,4,5}.

the final reference selection. However, when α reaches 0.95 and continues to increase, the final performance actually decreases to some extent, indicating the need to find a balance between the two for better results.

E Experimental Results for Different Prefix Lengths

In Figure 5, we conduct experiments with LLMRefLoc recalling different numbers of prefix tokens based on Llama-2-13b. We observe that longer prefix lengths do not bring additional performance improvements; on the contrary, they lead to a decrease in performance. Existing LLMs still perform better when generating shorter passages under constraints; longer passages introduce additional noise, resulting in decreased performance. However, overly short prefixes might also lack sufficient information, leading to an inability to accurately select the desired passage as a reference.

F Experimental Results of LLMs After General Fine-tuning

In Table 6, we compare the model performance in recalling references after supervised fine-tuning (Vicuna-1.3 and Vicuna-1.5 (Chiang et al., 2023)) and reinforcement learning from human feedback (Llama-2-chat). It is observed that the Vicuna models, after supervised fine-tuning, do not show further improvements; in fact, performance slightly declines. This indicates that the memorization of document knowledge primarily occurs during the pre-training phase, and further enhancement may require specific fine-tuning data and generation methods designed for recalling under constraints. In contrast, models fine-tuned with reinforcement learning from human feedback significantly drop in performance, unable to effectively recall reference under constraints. We note that models trained with reinforcement learning from human feedback often start their outputs with polite phrases like “Sure!”, which affects the model distribution and thus leads to failure.

G Experimental Results under Few-Shot Prompts

In Figure 6, we compare the experimental results of LLMRefLoc when incorporating few-shot prompts in the second stage of fine-grained passage recall based on Llama-2-13b. Using more sample prompts only brings partial improvements on the HotpotQA dataset, and the degree of improvement tends to stabilize when the number of samples increases from 3 to 5. On the NQ and TriviaQA datasets, no further improvements are observed, and a slight decline is even noted. There is still a need to explore more effective prompting methods.

H Experimental Results of Selecting Different Numbers of First Stage Documents

In Figure 7, we conduct experiments to compare the effect of selecting different numbers of first-stage documents (denoted as k) in LLMRefLoc when implemented based on Llama-2-13b. We observe that the impact of k on the final performance is not significant, as the necessary effective reference passages are usually contained within the first few

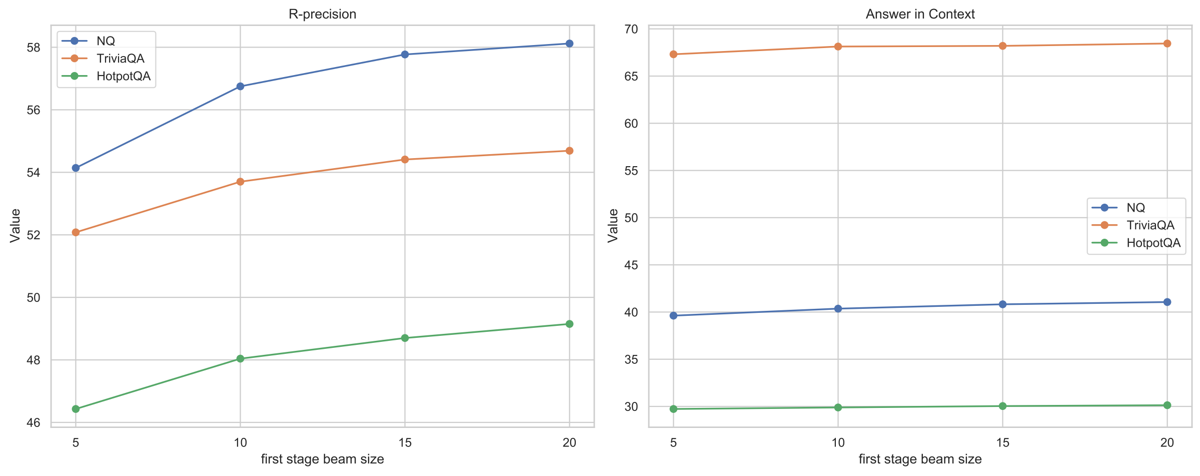


Figure 8: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for Llama-2-13b with different beam search sizes {4,8,16,32,64,128} set for the first stage.

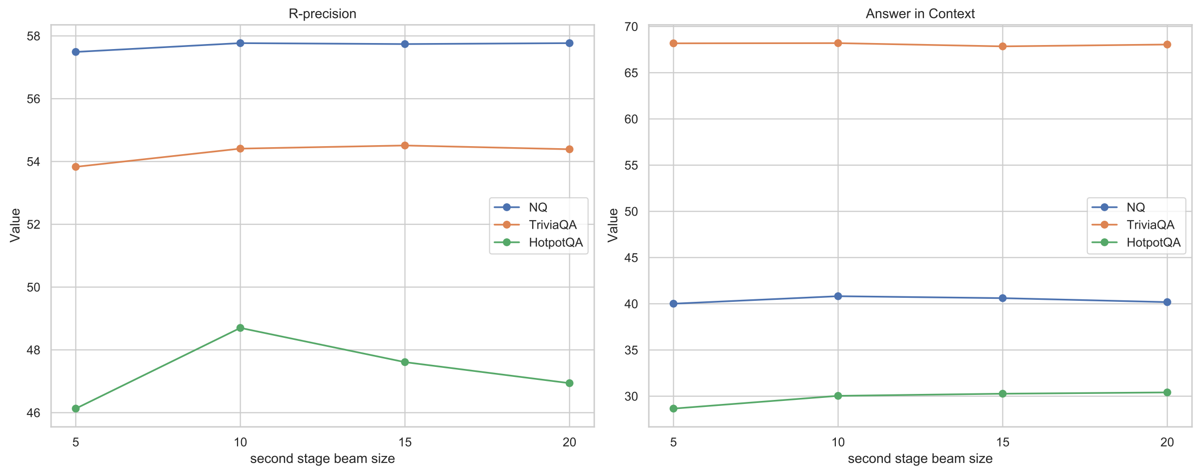


Figure 9: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for Llama-2-13b with different beam search sizes {4,8,16,32,64,128} set for the second stage.

documents. The suboptimal performance observed on the HotpotQA dataset when $k = 1$ can be attributed to the dataset requiring two documents to calculate R-Precision.

I Experimental Results with Different Beam Search Sizes

Figures 8 and 9 show the impact of setting different beam sizes in the first and second stages, respectively, in LLMRefLoc when implemented based on Llama-2-13b. For the first stage of recalling title identifiers, a larger beam size can achieve better page-level results, thereby slightly improving the effectiveness of the second stage of passage recall. However, in the second stage of fine-grained passage recall, the improvement brought by a larger beam size is not significant and may even lead to a slight decline, possibly due to the introduction of additional noise by a larger beam size.

J Case Study

In Tables 7 to 12, we present reference cases obtained using the Gold Standard, BM25, and the LLMRefLoc framework with Llama-2-13b on the NQ, TriviaQA, and HotpotQA datasets. By generating passages more aligned with the question, LLMRefLoc achieves results that contain the answer in Tables 7, 9, and 11. Table 8 showcases a biology question; although the passage recalled and located by LLMRefLoc does not contain the annotated answer, it provides a more detailed description of the location and process of pancreatic enzyme cleavage of peptide bonds. However, Tables 10 and 12 show instances where LLMRefLoc's recall failed. This is because merely generating a relevant prefix sometimes cannot ensure that the subsequent part will definitely contain the answer, leading to passages that are only broadly related. Ensuring the flexibility of recall and location while considering more subsequent information still requires further exploration. Nevertheless, we can also note that the references obtained by LLM recall are more natural and easier to read compared to those with predefined segmented beginnings. Finally, compared to the NQ dataset, questions in the TriviaQA and HotpotQA datasets are longer, allowing for more vocabulary overlap between questions and passages, which enhances the effectiveness of BM25 sparse retrieval.

Query: what is the legal age for marriage in australia
Gold Standard: Part II (s.10-21) deals with the marriageable age and the marriage of minors. In the original 1961 Act, marriageable age was set at 16 for females and 18 for males. However, under section 12 of the original 1961 Act a female 14 or 15 years or a male 16 or 17 years could apply to the court for permission to marry. The marriageable age was equalised in 1991 by the “Sex Discrimination Amendment Act 1991", which raised the marriageable age of females to 18 .
BM25: marriage. Before 2004, there was no definition of marriage in the 1961 Act, and instead the common law definition used in the English case “Hyde v Hyde" (1866) was considered supreme. Though s.46(1) of the Act required celebrants to explain the legal nature of marriage in Australia to a couple as “the union of a man and a woman to the exclusion of all others, voluntarily entered into for life", these words were descriptive or explanatory, rather than outlining what constituted a legally valid marriage in
LLMRefLoc: Marriage in Australia is regulated by the federal “Marriage Act 1961" (Cth) , which applies uniformly throughout Australia (including its external territories) to the exclusion of all state laws on the subject. Australian law recognises only monogamous marriages, being marriages of two people, including same-sex marriages, and does not recognise any other forms of union, including traditional Aboriginal marriages, polygamous marriages or concubinage. The marriage age for marriage in Australia is 18 years, but in “unusual and exceptional circumstances" a person aged 16 or 17 can marry with parental consent

Table 7: In the NQ dataset, the Gold Standard, BM25, and LLMRefLoc reference passage for the query “what is the legal age for marriage in Australia" are provided. The parts containing the answer are highlighted with a grey background.

Query: where does cleavage of the peptide bond by chymotrypsin occur
Gold Standard: “In vivo", chymotrypsin is a proteolytic enzyme (serine protease) acting in the digestive systems of many organisms. It facilitates the cleavage of peptide bonds by a hydrolysis reaction, which despite being thermodynamically favorable, occurs extremely slowly in the absence of a catalyst. The main substrates of chymotrypsin are peptide bonds in which the amino acid N-terminal to the bond is a tryptophan, tyrosine, phenylalanine, or leucine. Like many proteases, chymotrypsin also hydrolyses amide bonds “in vitro", a virtue that enabled the use of substrate analogs such as N-acetyl-L-phenylalanine p-nitrophenyl amide for enzyme assays.
BM25: 149, producing α-chymotrypsin (which is more active and stable than π-chymotrypsin). The resulting molecule is a three-polypeptide molecule interconnected via disulfide bonds. “In vivo", chymotrypsin is a proteolytic enzyme (serine protease) acting in the digestive systems of many organisms. It facilitates the cleavage of peptide bonds by a hydrolysis reaction, which despite being thermodynamically favorable, occurs extremely slowly in the absence of a catalyst. The main substrates of chymotrypsin are peptide bonds in which the amino acid N-terminal
LLMRefLoc: Chymotrypsin cleaves peptide bonds by attacking the unreactive carbonyl group with a powerful nucleophile, the serine 195 residue located in the active site of the enzyme, which briefly becomes covalently bonded to the substrate, forming an enzyme-substrate intermediate. Along with histidine 57 and aspartic acid 102, this serine residue constitutes the catalytic triad of the active site. These findings rely on inhibition assays and the study of the kinetics of cleavage of the aforementioned substrate, exploiting the fact that the en

Table 8: In the NQ dataset, the Gold Standard, BM25, and LLMRefLoc reference passage for the query “where does cleavage of the peptide bond by chymotrypsin occur" are provided. The parts containing the answer are highlighted with a grey background.

Query: Which expression is associated with the sinking of the HMS Birkenhead at Gansbaai near Cape Town, South Africa, in Febuary 1852?
Gold Standard: Only 193 of the estimated 643 people on board survived, and the soldiers' chivalry gave rise to the unofficial "women and children first" protocol when abandoning ship, while the "Birkenhead drill" of Rudyard Kipling's poem came to describe courage in face of hopeless circumstances.
BM25: HMS "Birkenhead, also referred to as HM Troopship "Birkenhead or Steam Frigate "Birkenhead", was one of the first iron-hulled ships built for the Royal Navy. She was designed as a steam frigate, but was converted to a troopship before being commissioned. She was wrecked on 26 February 1852, while transporting troops to Algoa Bay at Danger Point near Gansbaai, 87 miles (140 kilometres) from Cape Town in the Cape Colony. There were not enough serviceable lifeboats for all the passengers, and the soldiers
LLMRefLoc: The sinking of the "Birkenhead" is one of the earliest maritime disaster evacuations during which the concept of "women and children first" is known to have been applied."Women and children first" subsequently became standard procedure in relation to the evacuation of sinking ships, in fiction and in life. The term ""Birkenhead" drill" became defined as courageous behaviour in hopeless circumstances and appeared in Rudyard Kipling's 1893 tribute to the Royal Marines, "Soldier an' Sailor Too": To take your chance in the thick of a rush, with firing all about, Is nothing so bad when you've

Table 9: In the TriviaQA dataset, the Gold Standard, BM25, and LLMRefLoc reference passage for the query "Which expression is associated with the sinking of the HMS Birkenhead at Gansbaai near Cape Town, South Africa, in Febuary 1852?" are provided. The parts containing the answer are highlighted with a grey background.

Query: Who was the first winner of 'I'm A Celebrity, Get Me Out Of Here'?
Gold Standard: The first series of I'm a Celebrity...Get Me Out of Here! was broadcast on ITV from 25 August to 8 September 2002. Ant & Dec presented the main show on ITV, whilst Louise Loughman hosted the spin-off show "I'm a Celebrity...Get Me Out of Here! NOW!" on ITV2. The winner of this series was radio DJ Tony Blackburn .
BM25: The first series of I'm a Celebrity...Get Me Out of Here! was broadcast on ITV from 25 August to 8 September 2002. Ant & Dec presented the main show on ITV, whilst Louise Loughman hosted the spin-off show "I'm a Celebrity...Get Me Out of Here! NOW!" on ITV2. The winner of this series was radio DJ Tony Blackburn . The show began with 8 celebrity contestants. The contestants take part in daily trials to earn food All ratings are taken from the UK
LLMRefLoc: I'm a Celebrity...Get Me Out of Here! is a British reality TV series in which a number of celebrities live together in a jungle environment for a number of weeks, competing to be crowned "King" or "Queen of the Jungle". The show was originally created in the United Kingdom by the factual programmes department of ITV's then London franchise, London Weekend Television and developed by a team including James Allen, Natalka Znak, Brent Baker and Stewart Morris. The first episode aired on 25 August 2002 . It is now produced by ITV Studios and has been licensed globally to countries including the United States

Table 10: In the TriviaQA dataset, the Gold Standard, BM25, and LLMRefLoc reference passage for the query "Who was the first winner of 'I'm A Celebrity, Get Me Out Of Here'?" are provided. The parts containing the answer are highlighted with a grey background.

Query: 2014 S/S is the debut album of a South Korean boy group that was formed by who?
Gold Standard: 2014 S/S is the debut album of South Korean group WINNER. It was released on August 12, 2014 by the group's record label, YG Entertainment. The members were credited for writing the lyrics and composing the majority of the album's songs.
BM25: S is a South Korean project group consisting of three members: Kangta, Lee Ji-hoon and Shin Hye-sung. The group debuted in 2003, under the SM Entertainment label. After 11 years, they released and promoted another mini-album in 2014. In 2003, S released their first album "Fr.In.Cl", which stands for "Friends in Classic". In 2014, after more than a decade in hiatus, the group released their second mini-album "Autumn
LLMRefLoc: 2014 S/S is the debut album of South Korean group WINNER. It was released on August 12, 2014 by the group's record label, YG Entertainment. The members were credited for writing the lyrics and composing the majority of the album's songs. The members produced the majority of the music for this album themselves, with the help of other producers such as Choice 37, B.I., Airplay, and others. The album was highlighted for incorporating elements generally absent from K-pop releases, including hints of acoustic

Table 11: In the HotpotQA dataset, the Gold Standard, BM25, and LLMRefLoc reference passage for the query "2014 S/S is the debut album of a South Korean boy group that was formed by who?" are provided. The parts containing the answer are highlighted with a grey background.

Query: Who is the fictional head of a British Secret Service division and for which a one-time missionary was the inspiration for?
Gold Standard: Charles Fraser-Smith (26 January 1904 – 9 November 1992) was an author and one-time missionary who is widely credited as being the inspiration for Ian Fleming's James Bond quartermaster Q. During World War II, Fraser-Smith worked for the Ministry of Supply, fabricating equipment nicknamed "Q-devices" (after Q-ships) for SOE agents operating in occupied Europe. Prior to the war, Fraser-Smith had worked as a missionary in North Africa. After the war he purchased a dairy farm in Bratton Fleming, Devon, where he died in 1992.
BM25: Q is a fictional character in the James Bond films and film novelisations. Q (standing for Quartermaster), like M, is a job title rather than a name. He is the head of Q Branch (or later Q Division), the fictional research and development division of the British Secret Service. The use of letters as pseudonyms for senior officers in the British Secret Intelligence Service was started by its first director Captain Sir Mansfield George Smith-Cumming (1859-1923) who signed himself with a C written in
LLMRefLoc: Ian Fleming created the fictional character of James Bond as the central figure for his works. Bond is an intelligence officer in the Secret Intelligence Service, commonly known as MI6. Bond is known by his code number, 007, and was a Royal Naval Reserve Commander. Fleming based his fictional creation on a number of individuals he came across during his time in the Naval Intelligence Division and 30 Assault Unit during the Second World War, admitting that Bond "was a compound of all the secret agents and commando types I met during the war". Among those types were his brother, Peter, who had been involved in behind-the-lines operations in Norway and Greece during the war.

Table 12: In the HotpotQA dataset, the Gold Standard, BM25, and LLMRefLoc reference passage for the query "Who is the fictional head of a British Secret Service division and for which a one-time missionary was the inspiration for?" are provided. The parts containing the answer are highlighted with a grey background.