MS-RAG: Simple and Effective Multi-Semantic Retrieval-Augmented Generation

Xiaozhou You
Ant Group
Hangzhou, China
`youxz@antgroup.com`

Yahui Luo
Ant Group
Hangzhou, China
`haoshan.lyh@antgroup.com`

Lihong Gu
Ant Group
Hangzhou, China
`lihong.glh@antgroup.com`

Abstract

To alleviate the hallucination problem of large language model (LLM), retrieval-augmented generation (RAG) has been proposed and widely adopted. Due to the limitations in cross-chunk summarization task of naive RAG, graph-based RAG has emerged as a promising solution. However, a close study reveals several flaws in these works. First, most graph-based RAGs suffer from less efficient indexing process, which leads to information loss and expensive costs. Second, they heavily rely on LLM for retrieval thus inference slowly, which hinders their application in industry. To build a more efficient and effective RAG, we propose the multi-semantic RAG (MS-RAG). In this work, we combine knowledge graphs with dense vector to build a multi-semantic RAG. To be specific, (i) at indexing stage, we create multiple semantic-level indexes, including chunk-level, relation-level, and entity-level, to leverage the merits of dense vector and knowledge graph. (ii) at retrieval stage, unlike the previous LLM-empowered entity extraction, we propose a novel mix recall algorithm. Finally, we employ a multi-semantic rerank module to purify the results. Extensive experiments show that MS-RAG achieves superior performance. In terms of retrieval effect, MS-RAG achieves state-of-the-art performance, which is about 10%-30% improvement than the existing methods. In terms of question-answering effect, MS-RAG still achieves promising results with faster inference speed. More analysis and experiments are provided in Appendix.

1 Introduction

By integrating external knowledge, retrieval-augmented generation (RAG) effectively alleviates the hallucination problem of large language models (LLM) and has attracted widespread attention (Brown, 2020; Lewis et al., 2020; Fatehkia et al., 2024; Fang et al., 2024; Qian et al., 2024). Based on well-built knowledge index, RAG performs retrieval tailored to user needs. To further enhance the capacity, graph-based RAG was proposed and regarded as a promising solution. GraphRAG (Edge et al., 2024) utilizes LLM to extract entity and relation, and employs Leiden algorithm to build hierarchical graph communities. HippoRAG (Gutiérrez et al., 2024) is inspired by hippocampal indexing theory and uses Personalized PageRank algorithm (Haveliwala, 2002) for graph retrieval. Then, the following works have further improved graph-based RAG and promoted its application (Guo et al., 2024; Li et al., 2024; Yuan et al., 2024; Peng et al., 2024; Zhang et al., 2024).

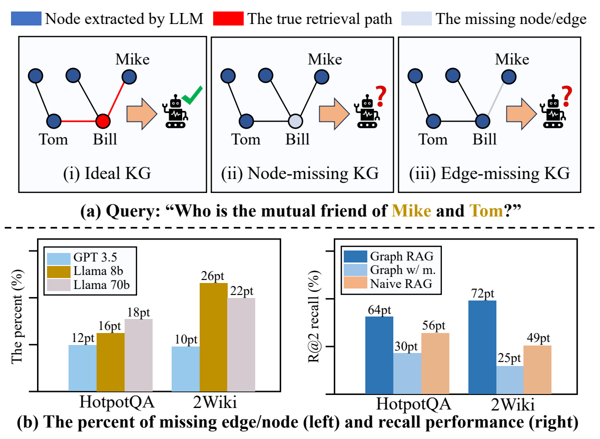

Figure 1: **(a)** On ideal knowledge graph (KG), the graph retrieval will find the answer "Bill" by locating the nodes ("Mike", "Tom"), and the corresponding edges, such as (i). However, if there are missing edges or nodes in KG, the retrieval will fail to find answer, such as (ii) and (iii). **(b)** We randomly selected 50 samples from the HotpotQA and 2WikiMultiHopQA and used different LLMs to construct KG. We found that 10-30% of the samples have missing node/edge. More importantly, when performing graph retrieval on missing node/edge KG (denoted graph w/ m.), the performance is even worse than Naive RAG.

However, by careful study, there are still several limitations. First, most graph-based RAG suffers

	Vector Search	Cross-Chunk	Multi-Semantic	Fast Inference	ReRank
Naive RAG	√	×	×	√	×
GraphRAG	×	√	×	×	×
HippoRAG	√	√	×	√	×
MS-RAG (ours)	√	√	√	√	√

Table 1: **The qualitative comparison between different RAGs.** Compared with existing methods, our proposed MS-RAG boasts several desired properties that contribute to building a more strong RAG. The "multi-semantic" refers to information that includes multiple semantic levels, such as chunk-level, entity-level, and relation-level.

from less efficient indexing process, which leads to information loss and expensive costs. Most of them inherently need LLM to build graph index, which is prone to mistakes. As shown in Figure 1, almost 10-30% of samples have missing node/edge. The performance of RAG with missing node/edge index is extremely poor, even worse than naive RAG. Besides, GraphRAG (Edge et al., 2024) needs to call LLM multiple times to build hierarchical communities, which brings expensive token costs and additional latency. Second, they heavily rely on LLM for retrieval thus inference slowly, which hinders their application in industry. Given a query, existing methods tend to use LLM to extract entities for following retrieval. The process not only has weak accuracy, but also makes inferences slowly.

To trackle it, we proposed a simple and effective multi-semantic RAG (MS-RAG). In this work, we combine knowledge graphs with dense embedding to build MS-RAG. First, at indexing stage, we established multiple semantic-level indexes, including chunk-level, relation-level, and entity-level, aimed to integrate the advantages of vector and graph. Some existing works rely on LLM for graph index construction, which easily leads to missing node/edge. We introduce chunk-level information in multi-semantic index, which not only helps to alleviate the negative of LLM mistakes, but also enhances semantic representation. In addition, we perform entity disambiguation to merge entities with same semantic. We also perform entity/relation summary to provide richer prompt to LLM.

Then, at retrieval stage, unlike the previous LLM-empowered entity extraction, we propose a novel mix recall algorithm to complementarily combine the advantages of graph recall and dense recall, then employ a novel multi-semantic rerank module to integrate different outputs. Given a query, we use vector search instead of LLM to extract entity/relation from our proposed multi-semantic index. Based on the recalled elements, we simultaneously perform graph search to recall other nearby entities and relations. Then, we propose an innovative multi-semantic rerank module to purify the final passage. The module first filters candidate passage by weighted voting, and then employ a lightweight LLM (Bai et al., 2023) to rerank.

Extensive experiments show MS-RAG achieves outstanding performance. We conduct experiments on public benckmarks like hotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022b). In terms of retrieval effect, compared with existing RAG methods, both in single-hop and multi-hop settings (Trivedi et al., 2022a), MS-RAG achieves state-of-the-art performance remarkably, improving retrieval accuracy by about 10%-30%. Besides, in terms of question-answering effect, compared with GraphRAG (Edge et al., 2024) and HippoRAG (Gutiérrez et al., 2024), MS-RAG achieves promising performance with faster inference speed.

2 Related Works

Large Language Model. In 2017, Transformer was proposed (Vaswani, 2017), which had a profound impact on subsequent work. Then, the BERT-like (Devlin et al., 2018) and GPT-like (Brown, 2020) models have demonstrated their capabilities on broader downstream tasks. In particular, ChatGPT (OpenAI, 2024) marked a significant milestone by accelerating the development of intelligent dialogue systems. The following works have further bolstered confidence in the pursuit of artificial general intelligence (AGI), such as Qwen (Bai et al., 2023) and Llama (AI@Meta, 2024). However, LLMs still face several challenges, including untimely data update and hallucination problem.

Retrieval-Augmented Generation. To alleviate the hallucination problem, RAG was proposed and widely used in LLM applications (Lewis et al., 2020). RAG mainly consists of indexing stage, retrieval stage and a generation stage. Recently,

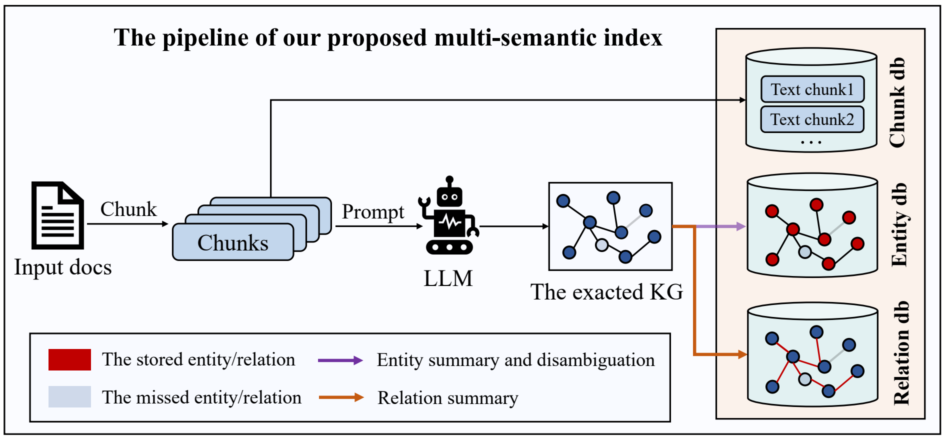
The pipeline of our proposed multi-semantic index
Input docs
Chunk
Chunks
Prompt
LLM
The exacted KG
Text chunk1
Text chunk2
Chunk db
Entity db
Relation db
🟥 The stored entity/relation
▨ The missed entity/relation
Entity summary and disambiguation
Relation summary

Figure 2: **The pipeline of our proposed multi-semantic index.** For the input documents, we establish a multi-semantic index, which contains three levels of information: chunk-level, entity-level and relation-level, to construct a more effective RAG index.

Many subsequent works have optimized the retrieval performance and generation effect of RAG. RAFT (Zhang et al., 2024) improves the robustness of LLM by training the model to ignore irrelevant documents, and introduces chain of thought (CoT) to improve the reasoning ability. And RankRAG (Yu et al., 2024) fine-tunes LLM to enable LLM to have both text generation and rerank capabilities.

Knowledge Graph for RAG. knowledge graph (Chen et al., 2020) are adopted in RAG due to their powerful ability of complex relationships understanding and cross-chunks summarization. GraphRAG (Edge et al., 2024) first applied graph to RAG, and effectively solved the query-summarization task by building a hierarchical graph community, and improved the generation quality. HippoRAG (Gutiérrez et al., 2024) combines knowledge graphs with Personalized PageRank algorithms to improve retrieval efficiency. However, these works face the problem of relying on LLM and low efficiency. And HybirdRAG (Sarmah et al., 2024) splices the results of VectorRAG and GraphRAG outputs together, in whch the output results of VectorRAG and GraphRAG are completely independent. Different from previous works, our proposed MS-RAG effectively improves retrieval performance and generation quality by introducing multi-semantic index.

3 Methods

3.1 Overview

To build a more efficient and effective retrieval-augmented generation system, we proposed MS-RAG. In this work, we integrate the merits of knowledge graphs and vector search.

As shown in Figure 2, it's the **indexing pipeline**. First, we chunk the input documents, and the chunked results are stored in chunk database. Then, we employ LLM to extract knowledge graph. The extracted entity stored in entity database, and the extracted relation stored in relation database. All chunks, entities, and relations are encoded by dense vector model. Besides, we also apply entity/relation summarization and disambiguation to enrich the prompt provided to LLM.

As shown in Figure 3, it's the **retrieval pipeline**. Based on the well-built multi-semantic index, we propose a novel mix recall algorithm to search related entities, relations, and chunks. Unlike previous methods (Gutiérrez et al., 2024; Guo et al., 2024), we first perform vector search to recall most relevant subject. Then, we recall some adjacent entities/relations and other similar chunks. Finally, the multi-semantic rerank module performs voting and rerank to purify final result.

3.2 Offline Indexing

Most existing works suffer from less efficient indexing process, which heavily rely on LLM to build graph index from input documents. To alleviate the negative impact of LLM mistakes, we propose multi-semantic index. To enrich prompt for LLM, we introduce summarization and disambiguation.

3.2.1 Multi-Semantic Index

The proposed multi-semantic index consists of three databases: entity, relation, and chunk. All

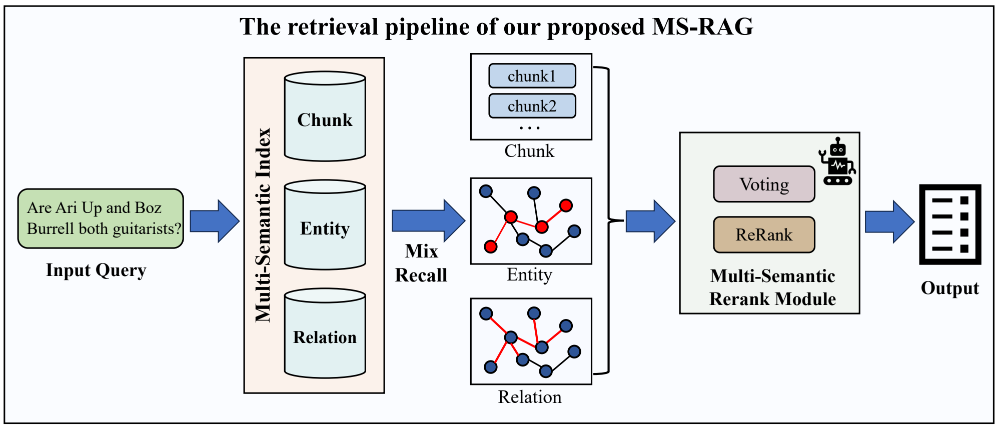
The retrieval pipeline of our proposed MS-RAG
Are Ari Up and Boz Burrell both guitarists?
Input Query
Multi-Semantic Index
Chunk
Entity
Relation
Mix Recall
chunk1
chunk2
...
Chunk
Entity
Relation
Voting
ReRank
Multi-Semantic Rerank Module
Output

Figure 3: **The retrieval pipeline of our proposed MS-RAG.** Based on the well-built multi-semantic index, we propose novel mix recall algorithm to complementarily combine the advantages of graph recall and dense recall. Then, we employ a novel multi-semantic rerank module to purify final outputs.

information of each database is encoded by a dense vector model.

Entity db. It mainly consists of some nouns and characters. For example, in the sentence "Tom loves basketball", "Tom" and "basketball" are entities encoded and stored in the database. In addition, the relation information of the entity and adjacent entities are also stored.

Relation db. It mainly consists of declarative sentences. For example, in the sentence "Tom loves basketball", entities ("Tom", "basketball") and relations ("play") are semantically concatenated together and then encoded by dense vector model.

Chunk db. It mainly consists of chunks from passage. The purpose of introducing naive chunk information in multi-semantic index is to alleviate the negative impact caused by LLM mistakes.

3.2.2 Summarization and Disambiguation

Entity/Relation Summarization: While building the graph index, we also designed prompts for LLM to generate summary for the extracted entity/relation, mainly based on involved context. We hope that the generated summary can serve as a helpful reference for the final outputs generated by LLM.

Entity Disambiguation: For some identical entities with different names, we decided to perform disambiguation, which we designed prompts to merge such case in indexing pipeline. For example, in a context, "Clarence Fred Gehrke" and "Fred Gehrke" refer to the same person.

3.3 Online Retrieval

To achieve more accurate and faster retrieval, we proposed two innovative components: mix recall algorithm and multi-semantic rerank module. The mix recall algorithm aims to better utilize the well-built multi-semantic index, which includes knowledge graph structure and chunk information. The multi-semantic rerank module enhances reasoning capabilities with slight inference delay. The process is illustrated in Figure 3.

3.3.1 Mix Recall Algorithm

The mix recall algorithm aims to complementarily combine the advantages of graph recall and dense recall, inspired by two aspects. First, knowledge graph provides an information structure with semantic associations. While dense recall struggles with complex and lengthy texts, it excels at recalling short entities and relations. Then, as shown in Figure 1, we introduced chunk recall to alleviate the negative impact of LLM errors. We also provide more comparison in Appendix C.

Graph Element Recall. It includes two stages: (i) entity/relation extraction: Previous works tend to use LLM for entity extraction (Gutiérrez et al., 2024; Edge et al., 2024). We argued that it's inefficient and time-consuming. In our work, we employ vector search instead of LLM to recall relevant subjects. We extracted λ_1 entities and λ_2 relations. (ii) semantic neighboring search: In knowledge graph, neighboring nodes and edges have associated semantics (Guo et al., 2024). For extracted relation, we recall its neighboring λ_h-hop relations. For extracted entity, we recall the neighboring entities and relations simultaneously.

Chunk Recall. Similar to naive RAG, we directly use vector tools to calculate similarity and take the top λ_3 chunks as the recall chunks. We introduced chunk recall to enrich our semantics and mitigate the impact of LLM mistakes.

3.3.2 Multi-semantic rerank module

The purpose of rerank is to evaluate the relation between the recall results and input query at semantic and logical level, and output more relevant ranking. To effectively integrate the results of multi-recall sources, we proposed multi-semantic reranking module, which mainly consists of two stages: voting and rerank.

Voting. We employ voting algorithm to integrate the results from multiple recall sources. Notably, the voting targets are the passage labels associated with entity, relation, or chunk. We argued that the passages appearing frequently in recall sources are more likely to be relevant. Based on this, we select the top λ_4 passages to advance to the next stage.

Rerank. Inspired by RankGPT (Sun et al., 2023), we design prompts and employ a lightweight LLM for rerank, with the purpose of using the reasoning ability of LLM to find relevant passage that can answer input query. In addition, in order to speed up inference speed, we prompt the large model to only output the sorted sequence number.

4 Experimental Setup

In this section, we introduce datasets, baselines, evaluation details, and implementation details about our experiments of our proposed MS-RAG.

4.1 Datasets

We conduct experiments primarily on three challenging multi-hop benchmarks, HotpotQA (Yang et al., 2018), MuSiQue (answerable) (Trivedi et al., 2022b) and 2WikiMultiHopQA (Ho et al., 2020). Following previous works (Press et al., 2023; Trivedi et al., 2022a; Gutiérrez et al., 2024), we selete 1,000 questions from each validation set. The description of datasets is shown below:

i) **HotpotQA.** It contains 113k Wikipedia QA pairs, where questions require searching and reasoning about multiple documents to answer. We selected about 9k passages for experiments.

ii) **MuSiQue.** It is a new multi-hop QA dataset containing about 25K 2-4 hop questions, using seed questions from 5 existing single-hop datasets. We selected about 12k passages for our experiments.

iii) **2WikiMultiHopQA.** It introduces evidence information containing reasoning paths for multi-hop questions, which contains structured and unstructured data. We selected about 6k passages for our experiments.

4.2 Baselines

In order to verify the effectiveness of MS-RAG, we compare against several strong and widely used retrieval methods. For retrieval performance, we compare with BGE-M3 (Chen et al., 2024), BM25 (Robertson and Walker, 1994), Contriever (Izacard et al., 2021), GTR (Ni et al., 2022) and ColBERTv2 (Santhanam et al., 2022). Besides, we compare with three recent LLM/RAG augmented baselines: HippoRAG (Gutiérrez et al., 2024), which introduces Personalized PageRank algorithm to enhance graph retrieval, and Propositionizer (Chen et al., 2023), which proposed a novel retrieval unit, and RAPTOR (Sarthi et al., 2024), which uses trees to integrate information from long documents. Besides, to verify the multi-step retrieval performance, we also compare with the multi-step retrieval method IRCoT (Trivedi et al., 2023). For question-answering performance, we compare with GraphRAG (Edge et al., 2024) and HippoRAG (Gutiérrez et al., 2024).

4.3 Evaluation Details

For the evaluation of retrieval performance, we use **top2 recall (R@2)** and **top5 recall (R@5)**, which refers to the proportion of correct answers in the top2 search results or in the top5 search results.

For the evaluation of question-answering performance, inspired by (Guo et al., 2024; Edge et al., 2024), we prompt LLM (Bai et al., 2023) to evaluate based on the following criteria:

i) **Correctness**: Does the answer correctly answer the question?

ii) **Diversity**: Does the answer provide a variety of perspectives on the question?

iii) **Comprehension**: Does the answer effectively help readers understand?

iv) **Overall**: Determine the best answer by combining the above three criteria.

4.4 Implementation Details

The corresponding hyper-parameters λ_1, λ_2, λ_3, λ_4, λ_h are set to 5, 10, 10, 5 and 4, respectively. We use GPT-3.5-turbo-1106 (OpenAI, 2024) with temperature of 0 as our LLM, and employ BGE-M3 (Chen et al., 2024) and Contriever (Izacard et al., 2021) as our retriever. The LLM used in proposed multi-semantic rerank module is Qwen-7B, and another LLM used to evaluate question-answer

	MuSiQue		2Wiki		HotpotQA		Average	
	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
BM25	32.3	41.2	51.8	61.9	55.4	72.2	46.5	58.4
Contriever	34.8	46.6	46.6	57.5	57.2	75.5	46.2	59.9
GTR	37.4	49.1	60.2	67.9	59.4	73.3	52.3	63.4
ColBERTv2	37.9	49.2	59.2	68.2	64.7	79.3	53.9	65.6
BGE-M3	38.2	51.7	62.1	70.2	69.5	82.4	56.6	68.1
RAPTOR	35.7	45.3	46.3	53.8	58.1	71.2	46.7	56.8
Proposition	37.6	49.3	56.4	63.1	58.7	71.1	50.9	61.2
HippoRAG	41.0	52.1	71.5	89.5	59.0	76.2	57.2	72.6
MS-RAG (Contriever)	43.2	54.3	77.8	90.9	77.6	86.2	66.2	77.1
MS-RAG (BGE-M3)	**47.3**	**56.4**	**80.1**	**91.2**	**79.4**	**88.5**	**68.9**	**78.7**

Table 2: **The single-step retrieval performance.** Our proposed MS-RAG achieves excellent performance, which outperforms all baselines on HotpotQA, MuSiQue and 2WikiMultiHopQA datasets.

	MuSiQue		2Wiki		HotpotQA		Average	
	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
IRCoT + BM25 (Default)	34.2	44.7	61.2	75.6	65.6	79.0	53.7	66.4
IRCoT + Contriever	39.1	52.2	51.6	63.8	65.9	81.6	52.2	65.9
IRCoT + ColBERTv2	41.7	53.7	64.1	74.4	67.9	82.0	57.9	70.0
IRCoT + HippoRAG	43.9	56.6	75.3	93.4	65.8	82.3	66.2	77.1
IRCoT + MS-RAG (Contriever)	45.7	57.6	80.4	93.8	78.4	88.7	68.2	79.3
IRCoT + MS-RAG (BGE-M3)	**47.8**	**59.2**	**85.0**	**95.2**	**82.7**	**91.2**	**71.8**	**81.9**

Table 3: **The multi-step retrieval performance.** Compared with other baselines, our proposed MS-RAG achieves state-of-the-art performance with significant improvements on multi-step setting.

performance is Qwen-32B (Bai et al., 2023). In Figure 1, the mentioned GraphRAG is (Edge et al., 2024), and the mentioned naive RAG is (Izacard et al., 2021), respectively.

5 Results

In this section, we will present the experimental results of our proposed MS-RAG, which includes single-step retrieval performance, multi-step retrieval performance, question-answering performance and other ablation experiments.

5.1 The single-step retrieval performance.

As shown in Table 2, MS-RAG outperforms all other methods with significant improvement, including general retrieval models such as ColBERTv2, GTR, BGE-M3, BM25 and Contriever, recent LLM-empowered baselines such as Propositionizer and RAPTOR, the current SOTA HippoRAG. Compared with previous best results, our MS-RAG obtains almost 20.5% R@2 and 8.4% R@5 improvement on average, respectively.

Compared with HippoRAG (Gutiérrez et al., 2024), the current SOTA graph-based RAG, our MS-RAG (Contriever) significantly improves performance. First, we introduce multi-semantic index, which organically combines the advantages of vector search and graph structure. We take advantage of the semantic structure of the graph and the accurate and fast characteristic of vector search to effectively improve the algorithm. At the indexing stage, we introduce chunk information to alleviate the problem of LLM errors. At the retrieval stage, we first use vector retrieval instead of LLM to perform entity extraction. In addition, we use neighby search instead of personal pagerank algorithm (Gutiérrez et al., 2024) to improve the efficiency and accuracy of graph retrieval. Finally, the proposed multi-semantic rerank module effectively integrates multi-semantic information. Im-

	Correctness	Diversity	Comprehension	Overall	Inference time
GraphRAG	27.7%	38.8%	45.7%	38.2%	4.12s
MS-RAG (ours)	72.3%	61.2%	54.3%	61.8%	0.76s
HippoRAG	38.8%	41.6%	44.3%	40.6%	0.88s
MS-RAG (ours)	61.2%	58.4%	55.7%	58.4%	0.76s

Table 4: **The QA performance on mixed datasets.** The paired comparison of question-answering effect between GraphRAG, HippoRAG and our MS-RAG, separately. Our proposed MS-RAG achieves better question-answering rating and obtains faster inference speed.

	R@2	R@5
MS-RAG	**79.4**	**88.5**
MS-RAG w/o multi-semantic	73.3	79.2
MS-RAG w/o rerank	76.1	84.9

Table 5: **The ablation experiments of MS-RAG on HotpotQA datasets.** The ablation experiments of different modules in proposed MS-RAG.

portantly, we demonstrate the importance of each of our proposed modules in Section 5.4 and the Appendix.

5.2 The multi-step retrieval performance.

As shown in Table 3, equipped with IRCoT, MS-RAG achieves more advanced performance than previous works on multi-step retrieval, including general retrieval models such as BM25, ColBERTv2, Contriever, and the current graph-based SOTA HippoRAG. Compared with previous best results, our MS-RAG obtains almost 8.5% R@2 and 8.4% R@5 improvement on average.

Compared with HippoRAG (Gutiérrez et al., 2024), the current state-of-the-art RAG, our MS-RAG improves R@2 metric from 66.2 to 71.8 (+5.6), improves R@5 metric from 77.1 to 81.9 (+4.8) on average. It directly reflects the innovation and effectiveness of our multi-semantic index in multi-step settings. In addition, the introduction of the multi-semantic rerank module can more effectively integrate different information.

Compared with IRCot + BM25 (Trivedi et al., 2023), the typical multi-step retrieval method, our MS-RAG demonstrated significant performance gains, which outperforms IRCoT + BM25 in all metrics and selected three datasets. On average, our MS-RAG improves R@2 metric from 53.7 to 71.8 (+18.1), improves R@5 metric from 66.4 to 77.1 (+10.7), which demonstrates the performance advantages of MS-RAG and the effectiveness of proposed novel modules.

5.3 The question-answering performance.

As shown in Table 4, it's the question-answering performance comparison between GraphRAG (Edge et al., 2024), HippoRAG (Gutiérrez et al., 2024) and our proposed MS-RAG. We select 100 questions from HotpotQA, 2wikiMultiHopQA and MuSiQue datasets. And we employ Qwen-32B (Bai et al., 2023) to judge the quality.

Our MS-RAG surpasses GraphRAG in all metrics, with 61.8% overall rating, 72.3% correctness, 61.2% diversity and 54.3% comprehension. For correctness, our MS-RAG surpasses GraphRAG from 27.7% to 72.3%. It is due to the reason that our proposed MS-RAG achieves more accurate retrieval and gives correct supporting facts. For diversity and comprehension, our MS-RAG surpasses GraphRAG from 38.8% to 61.2%, from 45.7% to 54.3%, respectively. We argued it's the result of our entity summarization and relation summarization. Besides, our MS-RAG achieves faster inference speed (0.76s vs 4.12s), more than 5 times improvement. It's because we reduced the number of LLM accesses and token consumption during our retrieval stage.

Compared with the current SOTA HippoRAG, MS-RAG surpasses it with 58.4% overall rating, 61.2% correctness, 58.4% diversity and 55.7% comprehension. For correctness, our MS-RAG imporves HippoRAG from 38.8% to 61.2%. It reflects the advanced retrieval performance of MS-RAG. For diversity and comprehension, our MS-RAG imporves HippoRAG from 41.6% to 58.4%, from 44.3% to 55.7%, respectively. It also proves the innovation and effectiveness of MS-RAG.

5.4 Ablations experiments

In this subsection, we will present the ablations Experiments of our MS-RAG, to evaluate the effect of each component, including ablation for modules in

	MuSiQue		2Wiki		HotpotQA		Average	
	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
Llama-3-8B-instruct	46.2	**57.3**	79.2	89.9	79.1	87.5	68.2	78.3
Llama-3-70B-instruct	46.8	57.2	79.5	90.6	79.3	88.1	68.5	78.6
REBEL	36.1	48.3	74.3	79.2	67.3	68.0	59.2	65.2
GPT-3.5-turbo	**47.3**	56.4	**80.1**	**91.2**	**79.4**	**88.5**	**68.9**	**78.7**

Table 6: **The ablation experiments of different LLMs in indexing stage on HotpotQA datasets.** By default, we choose GPT-3.5-turbo as the LLM to build our multi-semantic index.

	R@2	**R@5**
Mix recall algorithm (MR)	**79.4**	**88.5**
MR w/o chunk recall	77.1	86.2
MR w/o graph recall	70.6	79.1
MR w/ LLM extraction	73.8	82.2
MR w/ PPR	73.8	84.2

Table 7: **The ablation experiments of mix recall algorithm on HotpotQA datasets.** The ablation experiments of different methods in mix recall algorithm.

MS-RAG, ablation for different LLMs for indexing, and ablation for multi-semantic index.

5.4.1 Ablation for modules in MS-RAG.

As shown in Table 5, it's the ablation experiments of modules in MS-RAG on HotpotQA datasets. When we remove the multi-semantic index module and use only BGE-M3 (Chen et al., 2024) recall chunks, the R@2 metric decreases from 79.4 to 73.3 (-6.1), and the R@5 metric decreases from 88.5 to 79.2 (-9.3). It shows the effectiveness of integrating knowledge graphs and vectors.

And we remove the multi-semantic rerank module and take the most relevant output at different recalls (entity and relation first), the R@2 metric decreases from 79.4 to 76.1 (-3.3), and the the R@5 metric decreases from 88.5 to 84.9 (-3.6). It shows that the LLM-based voting and rerank algorithm we designed brings significant improvements.

5.4.2 Ablation for LLMs for indexing.

As shown in Table 6, it's the ablation experiments of different LLMs in our indexing stage on HotpotQA datasets. We try four different LLMs to build knowledge graph at indexing stage, including GPT-3.5-turbo (OpenAI, 2024), Llama-3-8B-instruct (AI@Meta, 2024), Llama-3-70B-instruct (AI@Meta, 2024), and REBEL (Huguet Cabot and Navigli, 2021). By default, we choose GPT-3.5-turbo as the LLM to build knowledge graph for our multi-semantic index, which achieves the best results overall. The results of Llama-3-70B-instruct is close to GPT-3.5-turbo. And REBEL is the worst performance. In the future, we will explore more LLMs to build a more complete knowledge graph.

5.4.3 Ablation for mix recall algorithm.

As shown in Table 7, it's the ablation experiments of mix recall algorithm on HotpotQA datasets. We try to remove chunk recall and graph recall separately to test the impact on performance. MR w/ LLM extraction means employs LLM to extract entity/relation instead of vector search. MR w/ PPR means employs PPR (Gutiérrez et al., 2024) to recall graph elements. When we remove any of the components, performance degrades, directly demonstrating the effectiveness of each component. Removing chunk recall, the R@2 accuracy drops by 2.3 and R@5 accuracy drops by 2.3, which indicates chunk recall plays a significant role in mitigating the negative effects of mistakes made by the LLM. Removing graph recall also get worse results. The attempts of using PPR and LLM extraction also demonstrated the innovativeness of the mix recall algorithm.

6 Conclusion

In this work, we propose multi-semantic RAG (MS-RAG) to build a more efficient and effective RAG. To be specific, (i) at indexing stage, we create multi-semantic index to leverage merits of dense vector and knowledge graph. (ii) at retrieval stage, we propose mix recall algorithm to capture texts at different semantic levels. Then we employ a multi-semantic rerank module to purify results. Extensive experiments show MS-RAG achieves superior performance. In the future, we will explore more effective retrieval methods to improve performance.

Limitations

Although MS-RAG achieves outstanding performance by introducing multi-semantic indexing and multi-semantic rerank modules, there are still some limitations. First, we need to explore ways to build knowledge graphs more effectively. Although our multi-semantic index effectively alleviates this problem, it is not enough. Second, we need to explore more advanced retrieval algorithms with MS-RAG to maximize the performance.

References

AI@Meta. 2024. Llama 3 model card.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2023. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert systems with applications*, 141:112948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. *arXiv preprint arXiv:2405.20978*.

Masoomali Fatehkia, Ji Kim Lucas, and Sanjay Chawla. 2024. T-rag: Lessons from the llm trenches. *arXiv preprint arXiv:2402.07483*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*.

Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv preprint arXiv:1705.03551*.

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

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.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2024. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *arXiv preprint arXiv:2410.08815*.

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

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2024. GPT-3.5 Turbo.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. *arXiv preprint arXiv:2402.09760*.

Stephen E. Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 232–241. ACM/Springer.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *arXiv preprint arXiv:2407.02485*.

Ye Yuan, Chengwu Liu, Jingyang Yuan, Gongbo Sun, Siqi Li, and Ming Zhang. 2024. A hybrid rag system with comprehensive enhancement on complex reasoning. *arXiv preprint arXiv:2408.05141*.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

A Contribution

In summary, our contributions are summarized as follows:

- **Lightweight Multi-Semantic Index with Lower LLM Dependency.** We built a multi-semantic index, including chunk-level, relation-level, and entity-level, with the goal of integrating the advantages of dense embedding and graph index. By introducing the chunk-level information, MS-RAG enrich our semantics and mitigate the negative impact of LLM mistakes effectively.
- **Novel Mix Recall Algorithm.** We propose a novel mix recall algorithm, which complementarily combines the advantages of graph recall and dense recall. It employs dense recall for more accurate short entity/relation extraction and graph structures for nearby recall logically.
- **Fast and Effective RAG Retrieval Pipeline.** By integrating vector search and knowledge graph, we achieve more accurate and faster RAG retrieval. The pipeline includes two innovative components: mix recall algorithm and multi-semantic rerank module. The multi-semantic rerank module enhances reasoning capabilities with slight inference delay.
- **The Outstanding Performance.** Extensive experiments show that MS-RAG achieves excellent performance. Compared with existing works, MS-RAG shows strong performance both in effect of retrieval and question-answering. In terms of retrieval effect, compared with existing RAG methods, both in single-hop and multi-hop settings (Trivedi et al., 2022a), MS-RAG achieves state-of-the-art performance remarkably, improving retrieval accuracy by about 10%-30%. Besides, in terms of question-answering effect, MS-RAG achieves promising performance with faster inference speed.

B Datasets details

Following IRCoT (Trivedi et al., 2023) and HippoRAG (Gutiérrez et al., 2024), we collect all candidate passages (including supporting and distractor passages) from our selected questions and form a retrieval corpus for each dataset. The specific information is show in Table 8.

	MuSiQue	2Wiki	HotpotQA
Passage	12k	6k	9k
Triplet	107k	51k	99k

Table 8: **The number of passages and extracted triplets on our selected datasets.**

C Discussion: Novel Recall Algorithm

The purpose of MS-RAG is to combine knowledge graph and vector retrieval to build a fast and effective RAG system and create an application that can be used in industry. First, we rethink the knowledge graph and vector model. Although GraphRAG (Edge et al., 2024) can solve the summarization problem well, it is difficult to achieve effective and fast retrieval, which makes it impossible to apply it on a large scale in industry. We believe that knowledge graph can bring the advantages of the basic data structure, which converts a long text into short entities and relations. The vector model may not perform well for complex and long text retrieval, but it can effectively recall short entities and relations. As shown in Figure 4, we discuss the the differences and connection between dense recall, graph recall and our mix recall. As shown in Algorithm 1, we show the mix recall algorithm pipeline, included the vector search on chunk index and nearby search on graph index.

Algorithm 1 Mix Recall pipeline

```
Require: input query $q$
Require: graph index $m_{graph}$
Require: chunk index $m_{chunk}$
 1: procedure
 2:     $rel, ent \leftarrow \textbf{VectorSearch}(q, m_{graph})$
 3:     $chunk \leftarrow \textbf{VectorSearch}(q, m_{chunk})$
 4:     $rel, ent \leftarrow \textbf{NearBy}(rel, ent, m_{graph})$
 5:     $p \leftarrow \textbf{Voting}(chunk, rel, ent)$
 6:     return $p \leftarrow \textbf{Rerank}(p)$
 7: end procedure
```

D Discussion: Distinct Advantage over Existing Solutions

We conduct discussion with existing HippoRAG, GraphRAG, and NaiveRAG to further emphasize the innovativeness of our framework.

HippoRAG. Compared with HippoRAG (Gutiérrez et al., 2024), our MS-RAG have significantly improved the retrieval performance

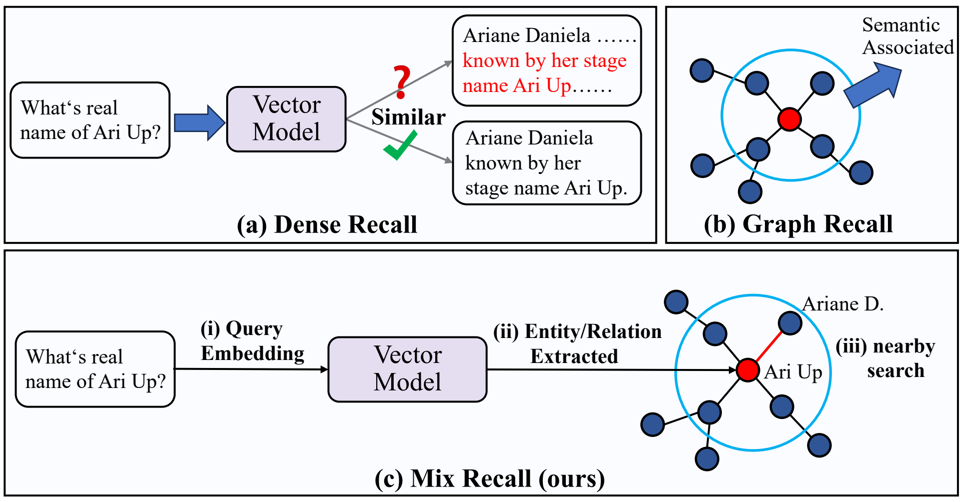
What's real name of Ari Up?
Vector Model
Similar
Ariane Daniela ...... known by her stage name Ari Up......
Ariane Daniela known by her stage name Ari Up.
(a) Dense Recall
Semantic Associated
(b) Graph Recall
What's real name of Ari Up?
(i) Query Embedding
Vector Model
(ii) Entity/Relation Extracted
Ariane D.
Ari Up
(iii) nearby search
(c) Mix Recall (ours)

Figure 4: **(a) Dense Recall:** dense recall struggles with complex and lengthy texts, but it excels at recalling short entities and relations. **(b) Graph Recall:** The KG provides the knowledge structure with semantic assoicitation, in which nearby nodes have explicit logical connections. **(b) Mix Recall (ours):** our proposed mix recall algorithm complementarily combines the advantages of graph recall and dense recall to achieve faster and more accurate retrieval.

	MuSiQue			2Wiki			HotpotQA		
	EM	F1	MRR	EM	F1	MRR	EM	F1	MRR
HippoRAG	19.2	29.8	57.4	46.6	59.5	79.4	41.8	55.0	68.2
MS-RAG (Contriever)	21.8	33.6	58.3	60.2	66.9	88.2	55.7	67.7	88.3
MS-RAG (BGE-M3)	**25.6**	**37.8**	**63.9**	**64.3**	**71.4**	**90.1**	**60.4**	**72.5**	**91.9**

Table 9: **The retrieval performance on F1, EM and MRR metric.** We conduct experiments to compare HippoRAG and MS-RAG on more metrics.

	NaturalQA	PopQA	TriviaQA
BM25	56.1	35.7	63.6
HippoRAG	70.3	49.2	72.8
HippoRAG	44.4	53.8	-
MS-RAG	**78.1**	**54.1**	**75.3**

Table 10: **The simpleQA experiments on NaturalQA, PopQA and TriviaQA datasets.** we add simpleQA experiments to further demonstrate the robustness.

	R@2	R@5
MS-RAG w/ qwen 7b	79.4	88.5
MS-RAG w/ Llama 8b	77.2	85.9
MS-RAG w/o rerank	76.1	84.9
HippoRAG	59.0	76.2

Table 11: **The ablation experiments of rerank module on HotpotQA datasets.** The ablation experiments of rerank LLMs in our proposed MS-RAG.

and question answering performance. (i) Indexing stage: The proposed multi-semantic index additionally introduces chunk index to mitigate the negative impact of LLM errors. (ii) Retrieval stage: We propose novel mix recall algorithm instead of PPR to complementarily combine the advantages of graph recall and dense recall, and propose a multi-semantic rerank module to give the model reasoning capabilities.

GraphRAG. Compared with GraphRAG (Edge et al., 2024), MS-RAG achieves better question-answering results while significantly reducing the number of LLM calls. We removed many unnecessary LLM calls, such as community clustering, community summarization, etc., which also accelerated inference speed.

Naive RAG. Compared with Naive RAG, our MS-RAG boost the performance with slight cost. Unlike vector search in naive RAG, we introduce the knowledge graph structure and propose an in-

Query	Is Ari Up a guitarists?
Support Passage	Ariane Daniela Forster, known by her stage name Ari Up, was a German vocalist best known as a member of The Slits.
Support Entity	Ari Up, Ariane Daniela Forster, German vocalist
Support Relation	(Ariane Daniela Forster, has stage name, Ari Up), (Ariane Daniela Forster, was, German vocalist), ...
Real Answer	Ari Up isn't a guitarist. The typical graph recall (HippoRAG), will locate the relations (Ariane Daniela Forster, was, German vocalist) based on the entities Ari Up and Ariane Daniela Forster, and thus get answer.
Challenge: Entity-missing	If important entities Ari Up or Ariane Daniela Forster are lost in the process of building the KG by LLM, graph recall will not be able to locate the answer.
Challenge: Relation-missing	If important relations (Ariane Daniela Forster, has stage name, Ari Up) or (Ariane Daniela Forster, was, German vocalist) are lost in the process of building the KG by LLM, graph recall will not be able to locate the answer.

Table 12: **The example of LLM errors for building the knowledge graph.** When graph-based RAG excute retrieval on the index with the missing node/edge, the performance will decrease.

	R@2	**R@5**
MS-RAG	79.4	88.5
MS-RAG w/o entity disa.	76.7	86.4

Table 13: **The ablation experiments of entity disambigguation on HotpotQA datasets.** The ablation experiments of entity disambigguation in our proposed MS-RAG.

Error Type	**Error Percent.(%)**
Incorrect/missing graph	26
Recall error	52
Rerank error	22

Table 14: **The error analysis on HotpotQA datasets.**

novative mix recall algorithm to better utilize our multi-semantic index.

E The Challenges for Constructing Knowledge Graph

To provide more details about LLM errors for building the knowledge graph, as shown in Table 12, we show an example to further analyze the challenge.

F The Experimental Results of F1, EM and MRR

To make it easier for readers to understand the effectiveness, we present the experiments between HippoRAG and MS-RAG in terms of F1 score, Exact Match (EM), and Mean Reciprocal Rank (MRR). As shown in Table 9, our proposed MS-RAG significantly outperforms the previous SOTA HippoRAG.

G The Experimental Results of SimpleQA tasks

To further demonstrate the robustness, we randomly selected 1,000 questions from NaturalQA (Kwiatkowski et al., 2019), 1,000 questionsfrom PopQA (Mallen et al., 2022), and 2,000 questions from TriviaQA (Joshi et al., 2017) to conduct experiments to demonstrate the performance of our model on the simpleQA task.

The experimental results are shown in Table 10. In terms of retrieval performance (passage recall@5), the proposed MS-RAG model significantly outperforms the dense model BGE-M3, the classic retriever BM25, and the baseline HippoRAG.

H Ablation of Rerank Module

As shown in Table 11, we conduct ablation experiments of rerank module on HotpotQA datasets. The experiments have shown that the qwen-7b rerank module can significantly improve performance. In addition, we also try Llama-3-8B-instruct as a chioce. When removing the rerank module, we also obtains better results than HippoRAG.

	GraphRAG	HippoRAG	MS-RAG (ours)
Indexing	extract graph, summarization, extract hierarchical community, ...	build graph	build graph
Retrieval	extract entity, summarization, community answers, ...	entity extraction	rerank

Table 15: **The comparison of LLM API calls between GraphRAG, HippoRAG and our MS-RAG in HotpotQA datasets.**

	chunk index	entity index	relation index	storage size
HippoRAG	×	√	√	2.1G
MS-RAG (ours)	√	√	√	2.4G

Table 16: **The comparison of storage costs between baseline HippoRAG and our MS-RAG in HotpotQA datasets.**

I Ablation of Entity Disambigguation

As shown in Table 13, we conduct ablation experiments of entity disambigguation on HotpotQA datasets. Our entity disambigguation achieves R@2 improvement of +2.7, R@5 improvement of +2.1, which demonstrates its effectiveness.

J Error Analysis

As shown in Table 14, we present error analsis on HotpotQA datasets. There are three error types. The main error type is recall error, which means the mix recall algorithm is not executed correctly. After voting, the correct subjects (chunks, relations/entiyies) doesn't enter the final rerank list. The second is incorrect/missing graph, which means there are missing entity/relation during the construction of KGs using LLM. The third is rerank error, which means the rerank model can's give right answer. The error analysis will help us improve the corresponding modules in the future.

K The Comparison of API Calls

As shown in Table 15, our MS-RAG achieves superior performance with very few LLM calls. Compared with GraphRAG, MS-RAG greatly reduces LLM calls. Compared with HippRAG, we achieve significant performance improvements with comparable LLM calls.

L The Comparison of Storage Costs

As show in Table 16, our multi-semantic index achieves better results only slight storage cost. Compared with HippoRAG, we only add an additional chunk index.

M The Experiments of Missing Edge/Node

We provide further details on the experiments concerning missing nodes/edges in Figure 1. Our evaluation focuses on the ratio of missing or incorrect triples. For example, if an passage contains 30 ground truth triples but only 24 are correctly extracted, the missing ratio is 20%. Since the HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) datasets do not have official annotations of ground truth triples, we manually annotated the ground truth triples to ensure annotation quality.

N LLM Prompts

Following HippoRAG (Gutiérrez et al., 2024), the knowledge graph is constructed in two stages, including entity extraction and relation extraction. The LLM prompts of relation extraction is shown in Figure 5, and the LLM prompts of entity extraction is shown in Figure 6. Besides, we also provide the LLM prompt of rerank module (Figure 7), entity disambiguation (Figure 8) and question-answering quality evaluation (Figure 9).

Instruction:

Your task is to construct an RDF (Resource Description Framework) graph from the given passages and named entity lists.

Respond with a JSON list of triples, with each triple representing a relationship in the RDF graph.

Pay attention to the following requirements:

- Each triple should contain at least one, but preferably two, of the named entities in the list for each passage.
- Clearly resolve pronouns to their specific names to maintain clarity.

Convert the paragraph into a JSON dict, it has a named entity list and a triple list.

One-Shot Demonstration:

Paragraph:

Radio City

Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.

Named_entities:

```
["Radio City", "India", "3 July 2001", "Hindi","English", "May 2008", "PlanetRadiocity.com"]
```

Triples:

```
[
    ["Radio City", "located in", "India"],
    ["Radio City", "is", "private FM radio station"],
    ["Radio City", "started on", "3 July 2001"],
    ["Radio City", "plays songs in", "Hindi"],
    ["Radio City", "plays songs in", "English"],
    ["Radio City", "forayed into", "New Media"],
    ["Radio City", "launched", "PlanetRadiocity.com"],
    ["PlanetRadiocity.com", "launched in", "May 2008"],
    ["PlanetRadiocity.com", "is", "music portal"],
    ["PlanetRadiocity.com", "offers", "news"],
    ["PlanetRadiocity.com", "offers", "videos"],
    ["PlanetRadiocity.com", "offers", "songs"]
]
```

Input:

Convert the paragraph into a JSON dict, it has a named entity list and a triple list.

Paragraph:

PASSAGE TO INDEX

Named_entities:

[entities]

Figure 5: **The LLM prompts of relation extraction.**

Instruction:

Your task is to extract named entities from the given paragraph. Respond with a JSON list of entities.

One-Shot Demonstration:

Paragraph:

Radio City

Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.

Named_entities:

["Radio City", "India", "3 July 2001", "Hindi","English", "May 2008", "PlanetRadiocity.com"]

Input:

Paragraph:

PASSAGE TO INDEX

Figure 6: **The LLM prompts of entity extraction.**

You are an excellent natural language processing expert. Your task is to search the text in textList based on the input query to find the text that can answer the user's query.

Requirements:

1. Return the first two texts that are helpful in answering the query.
2. Use # to concatenate the returned text numbers, such as 2#3.
3. Do not return correlation analysis, thinking process, and other irrelevant content.

One-shot Demonstration:

query: Are Ari Up and Boz Burrell both guitarists?

textList: [{"Text number": 0, "Text content": "Raymond "Boz" Burrell (1 August 1946 – 21 September 2006) was an English musician. Originally a vocalist and guitarist, Burrell is best known for his bass playing and work with the bands King Crimson and Bad Company. He died of a heart attack in Spain on 21 September 2006 aged 60."}, {"Text number": 1, "Text content": "Ariane Daniela Forster (17 January 1962 – 20 October 2010), known by her stage name Ari Up, was a German vocalist best known as a member of the English punk rock band The Slits."},{"Text number": 2, "Text content": "Muzzle is an alternative rock band formed in 1994 by Ryan Maxwell, Wesley Nelson, Burke Thomas, and Greg Collinsworth. They have released two albums with Reprise Records: "Betty Pickup" in 1996 and "Actual Size" in 1999."},{"Text number":"3","Text content":"Douglas Theodore "Doug" Pinnick (born September 3, 1950), sometimes stylized as dUg Pinnick or simply dUg, is an American musician best known as the bass guitarist, songwriter, and co-lead vocalist for the hard rock/progressive metal band King's amplifiers)."},{"Text number":"4","Text content":"Leslie West (born Leslie Weinstein; October 22, 1945) is an American rock guitarist, vocalist, and songwriter. He is best known as a founding member of the hard rock band Mountain."}]

Output: 0#1

Real query:

YOUR QUERY

textList:

YOUR TEXTLIST

Figure 7: **The LLM prompts of rerank module.**

Your task is to output different names for the same entity based on the input text.

One-Shot Demonstration:

input text:

Fred Gehrke. Clarence Fred Gehrke (April 24, 1918 \u2013 February 9, 2002) was an American football player and executive. He played in the National Football League (NFL) for the Cleveland / Los Angeles Rams, San Francisco 49ers and Chicago Cardinals from 1940 through 1950. To boost team morale, Gehrke designed and painted the Los Angeles Rams logo in 1948, which was the first painted on the helmets of an NFL team. He later served as the general manager of the Denver Broncos from 1977 through 1981. He is the great-grandfather of Miami Marlin Christian Yelich.

output:

["Fred Gehrke","Clarence Fred Gehrke"]

input text:

YOUR TEXT

Figure 8: **The LLM prompts of entity disambiguation.**

---Role---

You are an expert tasked with evaluating two answers to the same question based on three criteria: Correctness, Diversity, and Comprehension.

--Goal---

You will evaluate two answers to the same question based on three criteria: Correctness, Diversity, and Comprehension.

- Correctness: Does the answer correctly answer the question?
- Diversity: Does the answer provide a variety of perspectives on the question?
- Comprehension: Does the answer effectively help readers understand?

For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain why. Then, select an overall winner based on these three categories.

Here is the question:

{query}

Here is the groud-truth answer:

{groud-truth answer}

Here are the two answers:

Answer 1:

{answer1}

Answer 2:

{answer2}

Evaluate both answers using the three criteria listed above and provide detailed explanations for each criterion.

Output your evaluation in the following JSON format:

```
{{
"Correctness": {{ "Winner": "[Answer 1 or Answer 2]",
"Explanation": "[Provide explanation here]" }},
"Diversity": {{ "Winner": "[Answer 1 or Answer 2]",
"Explanation": "[Provide explanation here]" }},
"Comprehension": {{ "Winner": "[Answer 1 or Answer 2]",
"Explanation": "[Provide explanation here]" }},
"Overall Winner": {{ "Winner": "[Answer 1 or Answer 2]",
"Explanation": "[Summarize why this answer is the overall winner based on the three criteria]" }}
}}
```

Figure 9: **The LLM prompts of question-answering quality evaluation.**