MiniRAG: A Lightweight RAG system with Small Language Models

Tianyu Fan and Jingyuan Wang and Xubin Ren and Chao Huang
the University of Hong Kong
{tianyufan0504,jameswangjingyuan,xubinrencs,chaohuang75}@gmail.com

Abstract

The growing demand for efficient and lightweight Retrieval-Augmented Generation (RAG) systems has highlighted significant challenges when deploying Small Language Models (SLMs) in existing RAG frameworks. Current approaches face severe performance degradation due to SLMs' limited semantic understanding and text processing capabilities, creating barriers for widespread adoption in resource-constrained scenarios. To address these fundamental limitations, we present MiniRAG, a novel RAG system designed for simplicity and efficiency. MiniRAG introduces two key technical innovations: (1) a semantic-aware heterogeneous graph indexing mechanism that combines text chunks and named entities in a unified structure, reducing reliance on complex semantic understanding, and (2) a lightweight topology-enhanced retrieval approach that leverages graph structures for efficient knowledge discovery without requiring advanced language capabilities. Our extensive experiments demonstrate that MiniRAG achieves comparable performance to LLM-based methods even when using SLMs while requiring only 25% of the storage space. Additionally, we contribute a comprehensive benchmark dataset for evaluating lightweight RAG systems under realistic on-device scenarios with complex queries. We fully open-source our implementation and datasets at: https://github.com/HKUDS/MiniRAG.

1 Introduction

Recent advances in Retrieval-Augmented Generation (RAG) have significantly enhanced how language models access external knowledge, achieving strong performance in tasks like question answering and document synthesis (Fan et al., 2024). However, current RAG systems heavily rely on Large Language Models (LLMs) across all stages—from indexing and retrieval to response generation (Gao et al., 2023). This dependence imposes high computational costs and resource demands, hindering deployment in resource-constrained settings such as edge devices, privacy-sensitive applications, and real-time systems (Liu et al., 2024). Despite the growing need for efficient, lightweight solutions, existing RAG frameworks offer limited approaches to maintain robust performance under such practical constraints, revealing a critical gap between theoretical capabilities and real-world applicability.

The limitations of existing RAG systems become particularly apparent when attempting to utilize Small Language Models (SLMs) for resource-efficient deployment. While these models offer significant advantages in terms of computational efficiency and deployment flexibility, they face fundamental challenges in key RAG operations - from semantic understanding to effective information retrieval. Current RAG architectures (*e.g.*, LightRAG (Guo et al., 2024) and GraphRAG (Edge et al., 2024)), originally designed to leverage LLMs' sophisticated capabilities, fail to accommodate the inherent constraints of SLMs across multiple critical functions: sophisticated query interpretation, multi-step reasoning, semantic matching between queries and documents, and nuanced information synthesis. This architectural mismatch manifests in two significant ways: either severe performance degradation where accuracy drops, or complete system failure where certain advanced RAG frameworks become entirely inoperable when transitioning from LLMs to SLMs.

To address these fundamental challenges, we propose MiniRAG, a novel RAG system that reimagines the information retrieval and generation pipeline with a focus on simplicity and computational efficiency. Our design is motivated by three fundamental observations about SLMs: (1) while they struggle with sophisticated semantic understanding, they excel at pattern matching and localized text processing; (2) explicit structural in-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 37121–37144

formation can effectively compensate for limited semantic capabilities; and (3) decomposing complex RAG operations into simpler, well-defined steps can maintain system robustness without requiring advanced reasoning capabilities. These insights lead us to prioritize structural knowledge representation over semantic complexity, marking a significant departure from traditional LLM-centric RAG architectures.

Our MiniRAG introduces two key technical innovations that leverage these insights: (1) a semantic-aware heterogeneous graph indexing mechanism that systematically combines text chunks and named entities in a unified structure, reducing reliance on complex semantic understanding, and (2) a lightweight topology-enhanced retrieval approach that utilizes graph structures and heuristic search patterns for efficient knowledge discovery. These components work synergistically to enable robust RAG functionality even with limited model capabilities, fundamentally reimagining how RAG systems can operate within the constraints of SLMs while leveraging their strengths.

Through extensive experimentation across datasets and SLMs, we demonstrate MiniRAG's exceptional performance: compared to existing RAG systems, MiniRAG achieves 1.3-2.5× higher accuracy while using only 25% of the storage space. When transitioning from LLMs to SLMs, our system maintains remarkable robustness, with accuracy reduction ranging from merely 0.8% to 20% across different scenarios. Most notably, MiniRAG consistently achieves state-of-the-art performance across all evaluation settings, including tests on two comprehensive datasets with four different SLMs, while maintaining a lightweight footprint suitable for resource-constrained environments such as edge devices and privacy-sensitive applications. To facilitate further research in this direction, we also introduce LiHuaWorld, a comprehensive benchmark dataset specifically designed for evaluating lightweight RAG systems under realistic on-device scenarios such as personal communication and local document retrieval.

2 Related Works

Small Language Models (SLMs). The rise of SLMs is driven by the need for lightweight, efficient, and privacy-preserving AI that runs on edge devices, overcoming the computational and deployment constraints of LLMs (Liu et al., 2024; Team, 2024; Abdin et al., 2024; Hu et al., 2024). Notable SLMs—such as MiniCPM3-4B (Hu et al., 2024), phi-3.5-mini (Abdin et al., 2024), Llama-3.2-3B (Grattafiori et al., 2024), Qwen2.5-1.5B (Team, 2024), SmolLM-1.7B (Allal et al., 2024), and MobiLlama-1B (Thawakar et al., 2024)—achieve strong performance with far fewer parameters, offering fast inference, deployment flexibility, and enhanced privacy, making them well-suited for resource-constrained environments. Despite their success, the use of SLMs for RAG remains unexplored. This work addresses this gap by introducing a framework that enables SLMs to perform RAG effectively while preserving their efficiency and deployment advantages.

Retrieval-Augmented Generation (RAG). RAG enhances language model responses by retrieving relevant knowledge from external databases (Guo et al., 2024; Qian et al., 2024; Gao et al., 2024b), following a three-stage pipeline: indexing, retrieval, and generation. Two primary indexing strategies exist: (1) chunk-based methods that split text into retrievable segments, and (2) graph-based methods that encode knowledge as graphs. Chunk-based approaches—e.g., NaiveRAG (Mao et al., 2020), ChunkRAG (Singh et al., 2024), RQ-RAG (Chan et al., 2024)—focus on optimizing segmentation and retrieval, while graph-based methods—such as GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2024), and SubgraphRAG (Li et al., 2024a)—leverage graph structures for better comprehension and retrieval. However, most rely on either large context windows (Qian et al., 2024; Li et al., 2024a) or strong semantic understanding (Guo et al., 2024; Edge et al., 2024), limiting their suitability for small, lightweight language models and underscoring the need for more efficient RAG designs in resource-constrained settings.

3 The MiniRAG Framework

In this section, we present the detailed architecture of our proposed MiniRAG framework. As illustrated in Fig.1, MiniRAG consists of two key components: (1) heterogeneous graph indexing (Sec.3.1), which creates a semantic-aware knowledge representation, and (2) lightweight graph-based knowledge retrieval (Sec.3.2), which enables efficient and accurate information retrieval.

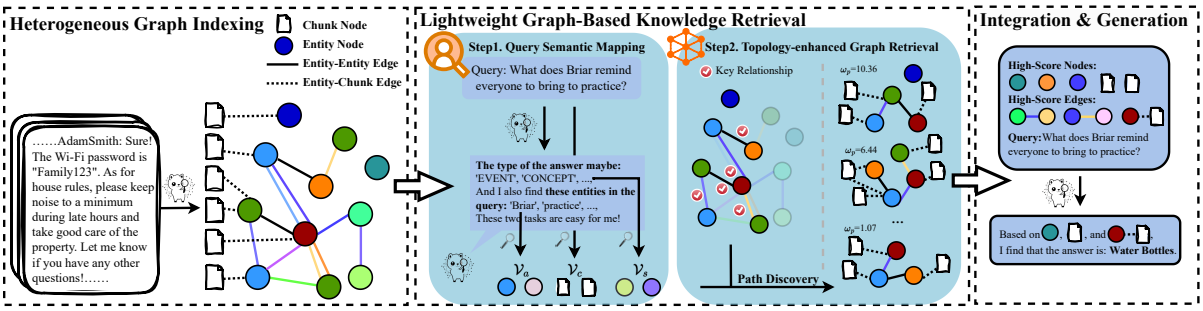

Figure 1: The MiniRAG employs a streamlined workflow built on the key components: heterogeneous graph indexing and lightweight graph-based knowledge retrieval. This architecture addresses the unique challenges faced by on-device RAG systems, optimizing for both efficiency and effectiveness.

3.1 Heterogeneous Graph Indexing with Small Language Models

In resource-constrained RAG systems, SLMs introduce significant operational constraints that impact their effectiveness. These limitations primarily manifest in two critical areas: (1) reduced capability to extract and understand complex entity relationships and subtle contextual connections from raw text, and (2) diminished capacity to effectively summarize large volumes of text and process retrieved information containing noise and irrelevant content. To address the challenges effectively, our MiniRAG aims to: **(1)** The indexing mechanism should extract the key relationships and contextual connections within the data, despite the SLMs' reduced entity understanding and summarization capacity. **(2)** The indexing approach should condense the retrieved content to its most query-relevant elements, thereby minimizing potential distractions or misleading information that could impair the SLM's capabilities in both summarization and effective denoising of the retrieved content.

To achieve the above goals, we propose a data indexing mechanism that generates a **Semantic-Aware Heterogeneous Graph**. This graph structure systematically incorporates both chunks and named entities extracted from the raw text, creating a rich semantic network that facilitates precise information retrieval. In the constructed heterogeneous graph, the nodes comprise two primary types: **Chunk Node** $\mathcal{V}_c$: Coherent segments of the original text corpus that preserve contextual integrity. **Entity Node** $\mathcal{V}_e$: The key semantic elements extracted from chunks include events, locations, temporal references, and domain-specific concepts that serve to anchor semantic understanding.

This dual-node design enables data chunks to directly participate in the retrieval stage, ensuring identification of the most contextually relevant content. This approach effectively mitigates information distortion that could arise from the limited summarization capabilities of the small language models. Within the heterogeneous graph, the connecting edges between nodes fall into two fundamental types: **Entity-Entity Connections** $\mathcal{E}_\alpha$: Links between named entities that capture semantic relationships, hierarchical structures, and temporal or spatial dependencies. **Entity-Chunk Connections** $\mathcal{E}_\beta$: Bridges between named entities and their corresponding contexts from which the entities are extracted, preserving contextual relevance and semantic coherence.

These connections and inter-dependencies are established through language models' semantic understanding capabilities. For example, when *indexing a document that plans a trip to the 2024 Paris Olympics*, the model might establish entity-entity connections between venue locations (Stade de France), event schedules (swimming finals), transportation options (Metro Line 13), and nearby attractions (Eiffel Tower), while creating entity-chunk connections linking these entities to relevant text segments discussing ticket availability, local accommodation reviews, and optimal travel routes.

To further facilitate the relational semantic understanding, we enhance each edge in the knowledge graph with semantic descriptions generated by language models. These descriptions provide explicit relationship context between connected nodes. Specifically, for each edge $e_\beta \in \mathcal{E}_\beta$ that connects an entity to its corresponding chunk, we employ the language model to generate a description d_{e_β} of the entity as supplementary information for this edge. This description provides detailed content about the entity and reflects the semantic

relationship between the extracted entity and the relevant chunk. With the textual description of the entity-chunk edge, it becomes a text-attributed edge $(e_\beta, d_{e_\beta}) \in \mathcal{E}_\beta$. In summary, the indexing process within our MiniRAG framework yields a heterogeneous graph $\mathcal{G}$ that encompasses both entity and chunk nodes with semantic-aware connections as follows:

$$\mathcal{D} = \mathcal{G} = (\{\mathcal{V}_c, \mathcal{V}_e\}, \{\mathcal{E}_\alpha, (e_\beta, d_{e_\beta}) \in \mathcal{E}_\beta\}) \quad (1)$$

3.2 Lightweight Graph-Based Knowledge Retrieval

In on-device RAG, the limitations of device computational capabilities and data privacy restrict the use of powerful models, such as LLMs and advanced text embedding models, necessitating reliance on smaller alternatives. These smaller models often struggle to capture the precise semantic nuances within lengthy texts, complicating accurate matching. To tackle these challenges, it is essential to: (1) **Reduce the complexity of input content** for generation, ensuring that semantic information is clear and concise; (2) **Shorten the length of input content** for smaller language models, facilitating improved comprehension and retrieval accuracy. Additionally, employing effective graph indexing structures can help mitigate performance deficiencies in semantic matching, thereby enhancing the overall retrieval process.

In MiniRAG, we propose a **Graph-based Knowledge Retrieval** mechanism that effectively leverages the semantic-aware heterogeneous graph $\mathcal{G}$ constructed during the indexing phase, in conjunction with lightweight text embeddings, to achieve efficient knowledge retrieval. By employing a graph-based search design, we aim to ease the burden on precise semantic matching with language models. This approach facilitates the acquisition of rich and accurate textual content at a low computational cost, thereby enhancing the ability of language models to generate precise responses.

3.2.1 Query Semantic Mapping

In the retrieval phase, the primary goal for a user-input query q is to identify elements related to the query (e.g., named entities, text chunks) from the constructed index data, thereby aiding the model in generating accurate responses. To achieve this, it is essential to first parse the query and align it with the index data. Some prior RAG methods utilize LLMs to expand or decompose the query into fine-grained queries (Chan et al., 2024; Edge et al., 2024; Guo et al., 2024), enhancing the match between the query and the index data. However, this process relies on LLMs to extract high-quality abstract information from the query, which poses challenges for smaller language models. Therefore, in the retrieval process of MiniRAG, we leverage entity extraction—a relatively simple and effective task for small language models—to facilitate the decomposition and mapping of the query q to our graph-based indexing data (i.e., the semantic-aware heterogeneous graph $\mathcal{G}$).

For a given q, our approach begins with a two-stage entity processing pipeline. First, we employ the small language model to extract relevant entities $\mathcal{V}_q$ from q while simultaneously predicting their potential types $\mathcal{T}$(*e.g.*, event, location, person) that may directly contribute to the query's answer. Following this, we leverage a lightweight sentence embedding model to evaluate semantic similarities across all entity nodes $\mathcal{V}_e$ in the constructed graph $\mathcal{G} = \{\mathcal{V}_c, \mathcal{V}_e\}$, examining various text corpora (*i.e.*, entity names, chunk content) to enable effective node retrieval and grounding.

Query-Driven Reasoning Path Discovery. Within our semantic-aware heterogeneous graph $\mathcal{G} = \{\mathcal{V}_c, \mathcal{V}_e\}$, MiniRAG enables more effective and interpretable reasoning by constructing query-guided reasoning paths. Given an input query q, the model aims to uncover meaningful and semantically coherent paths that connect relevant knowledge across the graph. To achieve this, it identifies pertinent text chunks by evaluating: (1) the semantic relevance between the query and entity nodes, ensuring that selected entities are contextually aligned with the user's intent; and (2) the structural coherence among relationships $\mathcal{V}_e$–$\mathcal{V}_e$ and $\mathcal{V}_e$–$\mathcal{V}_c$, which preserves the underlying topology and relational semantics of the graph. This dual-objective framework is designed to simultaneously maximize the semantic alignment between the query q and the entity nodes $\mathcal{V}_e$, while maintaining the integrity of relational dependencies. By balancing these objectives, MiniRAG effectively captures complex, multi-hop reasoning chains that span both semantic and structural dimensions of the heterogeneous graph.

The systematic query-relevant reasoning path discovery procedure consists of the following key steps: **Initial Entity Identification** ($\hat{\mathcal{V}}_s$): We locate high-confidence starting points by matching query entities with graph nodes, establishing reliable en-

try points for path exploration. **Answer-Aware Entity Selection** ($\hat{\mathcal{V}}_a$): Leveraging predicted answer types $\mathcal{T}$, we identify candidate answer nodes from the starting set, enabling type-guided reasoning. **Context-Rich Path Formation** ($\hat{\mathcal{V}}_c$): We enrich reasoning paths by incorporating relevant text chunks, creating comprehensive evidence chains that connect query entities to potential answers.

This lightweight framework maintains high efficiency while ensuring semantic accuracy, making it particularly suitable for edge computing scenarios. The subsequent section details our search algorithm for further refining these reasoning paths through importance-based ranking.

3.2.2 Topology-Enhanced Graph Retrieval

To address the limitations of SLMs in knowledge retrieval, particularly their poor semantic understanding, which introduces noise due to difficulty capturing nuanced meanings, context, and complex entity relationships, we propose a **topology-enhanced retrieval approach**. Our method effectively combines semantic and structural information from heterogeneous graphs through a two-stage process that synergistically leverages embedding-based similarities and graph topology.

The process begins with embedding-based similarity search to identify seed entities ($\hat{\mathcal{V}}_s$, $\hat{\mathcal{V}}_a$) through semantic matching, followed by a topology-enhanced discovery phase that leverages the heterogeneous graph structure $\mathcal{G}$ to discover relevant reasoning paths. By integrating entity-specific relevance scores, structural importance metrics, and path connectivity patterns, our approach achieves superior precision in knowledge retrieval while maintaining computational efficiency, ultimately enabling more accurate and interpretable reasoning paths for enhanced generation tasks.

Key Relationship Identification: We first identify high-quality entity-entity connections within graph $\mathcal{G}$ that are relevant to query q through node-edge interactions. In the entity-entity connections $\mathcal{E}_\alpha$, we define an edge as highly relevant if it connects a starting node $\hat{v}_s \in \hat{\mathcal{V}}_s$ to an answer node $\hat{v}_a \in \hat{\mathcal{V}}_a$ along their shortest path. For efficient extraction, we focus on edges proximate to starting or answer nodes. The relevance scoring function $\omega_e(\cdot)$ for each edge $e_\alpha \in \mathcal{E}_\alpha$ is formally defined:

$$\omega_e(e) = \sum_{\hat{v} \in \{\hat{\mathcal{V}}_s, \hat{\mathcal{V}}_a\}} \text{count}(\hat{v}, \hat{\mathcal{G}}_{e,k}) \qquad (2)$$

where $\hat{\mathcal{G}}_{e,k}$ denotes the k-hop subgraph centered at edge e, encompassing all nodes and edges reachable within k steps from either endpoint and $\text{count}(v, \mathcal{G})$ serves as a binary indicator function that returns 1 if node v appears in $\mathcal{G}$ and 0 otherwise. Based on the computed relevance scores ω_e, we construct the key relationships set $\hat{\mathcal{E}}_\alpha$ by selecting the top-ranked edges.

Query-Guided Path Discovery: To systematically discover logically relevant information within our knowledge graph structure, we identify and extract significant paths that serve as meaningful reasoning chains. A reasoning path starts from a carefully selected seed node and progressively advances toward potential answer nodes while maximizing the incorporation of previously extracted key relationships. For each candidate starting node $\hat{v}_s$ in our graph, we comprehensively define the potential reasoning path set $\mathcal{P}\hat{v}_s$ as the collection of all possible acyclic paths of length n originating from $\hat{v}_s$. For each identified query entity $v_q \in \mathcal{V}_q$, we systematically evaluate these potential paths using a sophisticated entity-conditioned score function $\omega_p(\cdot)$ that quantifies both the overall path importance and query relevance through multiple dimensions:

$$\omega_p(p \mid v_q) = \omega_v(\hat{v}_s \mid v_q) \cdot (1 + \sum_{e \in (p \wedge \hat{\mathcal{E}}_\alpha)} \omega_e(e) + \sum_{v \in (p \wedge \hat{\mathcal{V}}_a)} \text{count}(v, p)). \qquad (3)$$

The scoring components in our path discovery framework are defined as follows: $\omega_v(\hat{v}_s \mid v_q)$ measures the semantic similarity between starting node $\hat{v}_s$ and query entity v_q using cosine similarity of their respective embeddings in the vector space, while $\text{count}(v, p)$ serves as a binary indicator function that returns 1 if node v appears in path p and 0 otherwise. For each query entity and starting node pair in the graph, we systematically rank all potential paths according to their computed importance scores and construct the final comprehensive set of reasoning paths $\mathcal{P}_q$ by carefully selecting the top-k highest-scoring paths from each ranking list for subsequent steps.

Retrieval of Query-Relevant Text Chunks: Building upon our indexing structure from Section 3.1, each entity node maintains connections with its source text chunk through entity-chunk inter-dependencies. These text chunks exist as nodes in our indexing graph, connected via text-

attributed edges $(e_\beta, d_{e_\beta}) \in \mathcal{E}_\beta$. By traversing these connections, we collect all chunk nodes $\mathcal{V}_c^q$ that are linked to entity nodes present in any reasoning path $p \in \mathcal{P}_q$. **Step 1: Candidate Filtering.** We first systematically filter the candidates to focus on the intersection $\hat{\mathcal{V}}_c \wedge \mathcal{V}_c^q$ to ensure coverage of relevant information. **Step 2: Similarity Computation.** For each candidate chunk in the intersection, we carefully calculate the semantic similarity between the input query and the concatenated content, which combines both the chunk and its associated edge descriptions. **Step 3: Ranking and Selection.** Finally, we rank these filtered chunks according to their computed relevance scores and select the top-scoring candidates to form the final optimized set $\hat{\mathcal{V}}_c^q$ for subsequent reasoning.

Integration for Augmented Generation: Building upon our proposed topology-enhanced retrieval mechanism and multi-stage filtering pipeline, we systematically extract and refine two essential components of query-relevant knowledge: (1) *Essential relationships* $\hat{\mathcal{E}}_\alpha$, which represent the most semantically meaningful and structurally significant connections between key entities, effectively capturing the relational dependencies and higher-order patterns critical to understanding the query context; and (2) *Optimal text chunks* $\hat{\mathcal{V}}_c^q$, which are carefully selected segments of evidence containing rich contextual details and factual support necessary for generating answers. These components are not only highly relevant to the input query but are also purged of noise and redundancy through our filtering stages, ensuring both precision and efficiency. By systematically integrating these retrieved components with the previously grounded answer nodes $\hat{\mathcal{V}}_a$, we construct the comprehensive and well-structured input representation for the final augmented generation process.

4 Evaluation

Through the novel design of MiniRAG, we enable efficient on-device RAG systems without relying on large models, preserving data privacy while maintaining robust performance. Our evaluation addresses three key research questions (RQs):

RQ1: Performance. How does MiniRAG perform against state-of-the-art alternatives?

RQ2: Component Analysis. What is the contribution of key components to MiniRAG's overall effectiveness?

RQ3: Efficiency. What is MiniRAG's storage efficiency and scalability?

4.1 Experimental Settings

Datasets. Evaluating on-device RAG requires focusing on its unique context—privacy-preserving, real-time access to personal, lightweight content rather than large-scale document processing. Accordingly, we use two datasets reflecting core on-device use cases: (1) **Synthetic Personal Communication Data**. We create LiHuaWorld, a year-long, privacy-preserving dataset that realistically captures daily interactions (e.g., necessities, socializing, work/study, scheduling, shopping) with natural communication patterns and temporal-contextual coherence, suitable for "Instant Messaging" and "Personal Content" evaluation; and (2) **Short Documents**. MultiHop-RAG (Tang and Yang, 2024) based on news articles, designed to assess retrieval efficiency and accuracy across multiple short, locally stored files, mirroring real-world "Local Short Documents" scenarios in on-device settings.

Evaluation Metrics. We employ two metrics to assess the quality and reliability of responses. • **Accuracy** (acc): Measures the consistency between the RAG system's response and the expected answer. For instance, given the query "What does Briar remind everyone to bring to practice?" with the expected answer "bottle", semantically equivalent responses like "water bottle" are considered correct. • **Error Rate** (err): Captures instances where the RAG system provides incorrect information without recognizing its mistake. For example, if the system responds with "yoga mat" , it would count toward the error rate. Judgments are made via GPT-4o. To ensure robustness, we repeated each evaluation three times with order randomization, and reporting means and variance in results.

RAPTOR (Santhanam et al., 2021) use a tree-like structure to retrive information. HippoRAGv2 (Gutiérrez et al., 2025) and GraphRAG (Edge et al., 2024) constructs a graph-based index using LLMs. LightRAG (Guo et al., 2024) employs a dual-level retrieval architecture with knowledge graphs. We apply these methods in an end-to-end manner, using a single model to construct the index, generate search queries, and answer questions. Following the setup of the adopted methods, we do not employ a reranker.

For more datasets results, hyperparameter analysis, more details of datasets, baselines, and implementation details, please refer to Appendix A and B.

Table 1: Performance evaluation using accuracy (acc) and error (err) rates, measured as percentages (%). Higher accuracy and lower error rates indicate better RAG performance. Results compare various baseline methods against our MiniRAG across multiple datasets. Bold values indicate best performance, while "/" denotes cases where methods failed to generate effective responses.

LiHuaWorld	BM25		RAPTOR		HippoRAGv2		GraphRAG		LightRAG		MiniRAG	
	acc↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓
Phi-3.5-mini-instruct	$41.60_{\pm 3.61}$%	$21.19_{\pm 1.88}$%	$37.23_{\pm 3.28}$%	$22.21_{\pm 2.58}$%	$36.77_{\pm 2.98}$%	$27.14_{\pm 3.14}$%	/	/	$40.03_{\pm 4.40}$%	$25.59_{\pm 2.83}$%	**$50.71_{\pm 4.71}$**%	$24.65_{\pm 2.67}$%
GLM-Edge-1.5B-Chat	$42.39_{\pm 4.08}$%	$24.80_{\pm 2.20}$%	$43.20_{\pm 2.19}$%	$24.17_{\pm 2.82}$%	$33.92_{\pm 3.64}$%	$21.86_{\pm 3.11}$%	/	/	$34.85_{\pm 4.71}$%	$24.65_{\pm 3.45}$%	**$51.65_{\pm 4.08}$**%	$26.84_{\pm 2.98}$%
Qwen2.5-3B-Instruct	$40.97_{\pm 5.02}$%	$24.33_{\pm 3.92}$%	$37.49_{\pm 4.13}$%	$21.73_{\pm 3.43}$%	$36.58_{\pm 2.66}$%	$26.77_{\pm 3.43}$%	/	/	$41.76_{\pm 3.77}$%	$30.77_{\pm 2.83}$%	**$46.78_{\pm 4.55}$**%	$24.33_{\pm 3.77}$%
MiniCPM3-4B	$41.44_{\pm 4.24}$%	$25.27_{\pm 2.35}$%	$41.76_{\pm 3.82}$%	$27.69_{\pm 3.91}$%	$35.79_{\pm 3.82}$%	$25.83_{\pm 2.51}$%	/	/	$36.26_{\pm 4.40}$%	$24.49_{\pm 2.04}$%	**$51.18_{\pm 4.08}$**%	$22.29_{\pm 2.51}$%
gpt-4o-mini	$42.54_{\pm 3.30}$%	$20.88_{\pm 2.04}$%	$52.62_{\pm 4.40}$%	$19.37_{\pm 2.74}$%	**$66.28_{\pm 2.06}$**%	$17.65_{\pm 1.26}$%	$37.68_{\pm 2.20}$%	$38.30_{\pm 1.26}$%	$55.89_{\pm 3.45}$%	$18.52_{\pm 6.12}$%	$52.28_{\pm 3.14}$%	$21.35_{\pm 3.77}$%
MultiHop-RAG	BM25		RAPTOR		HippoRAGv2		GraphRAG		LightRAG		MiniRAG	
	acc↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓
Phi-3.5-mini-instruct	$38.03_{\pm 2.82}$%	$26.37_{\pm 1.33}$%	$40.15_{\pm 3.17}$%	$28.63_{\pm 2.57}$%	$30.58_{\pm 2.36}$%	$23.52_{\pm 3.04}$%	/	/	$26.76_{\pm 3.01}$%	$10.84_{\pm 1.21}$%	**$52.74_{\pm 3.48}$**%	$30.79_{\pm 1.60}$%
GLM-Edge-1.5B-Chat	$41.55_{\pm 2.66}$%	$22.77_{\pm 1.60}$%	$42.59_{\pm 2.49}$%	$32.32_{\pm 3.22}$%	$38.51_{\pm 3.77}$%	$32.58_{\pm 3.88}$%	/	/	/	/	**$52.39_{\pm 3.56}$**%	$21.24_{\pm 1.92}$%
Qwen2.5-3B-Instruct	$38.58_{\pm 2.19}$%	$23.71_{\pm 2.78}$%	$39.50_{\pm 2.37}$%	$28.18_{\pm 2.10}$%	$38.56_{\pm 2.94}$%	$29.84_{\pm 3.37}$%	/	/	$21.87_{\pm 4.03}$%	$14.71_{\pm 3.36}$%	**$50.94_{\pm 4.58}$**%	$35.49_{\pm 3.87}$%
MiniCPM3-4B	$41.90_{\pm 3.05}$%	$21.28_{\pm 2.46}$%	$44.78_{\pm 3.38}$%	$34.29_{\pm 3.06}$%	$37.12_{\pm 2.76}$%	$32.45_{\pm 2.87}$%	/	/	$19.99_{\pm 2.39}$%	$12.44_{\pm 1.21}$%	**$50.23_{\pm 4.26}$**%	$29.62_{\pm 2.78}$%
gpt-4o-mini	$39.79_{\pm 1.96}$%	$23.51_{\pm 2.86}$%	$56.42_{\pm 3.06}$%	$18.77_{\pm 1.94}$%	**$72.46_{\pm 4.95}$**%	$16.88_{\pm 2.45}$%	$59.27_{\pm 4.03}$%	$15.49_{\pm 1.41}$%	$66.55_{\pm 4.97}$%	$18.04_{\pm 3.29}$%	$68.54_{\pm 3.95}$%	$16.94_{\pm 3.25}$%
2wikimultihopqa	BM25		RAPTOR		HippoRAGv2		GraphRAG		LightRAG		MiniRAG	
	acc↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓	*acc*↑	*err*↓
Phi-3.5-mini-instruct	$44.42_{\pm 4.28}$%	$31.79_{\pm 3.30}$%	$44.86_{\pm 4.52}$%	$38.26_{\pm 3.95}$%	$36.65_{\pm 4.21}$%	$26.24_{\pm 3.12}$%	/	/	$24.21_{\pm 3.29}$%	$12.34_{\pm 2.84}$%	**$45.91_{\pm 6.02}$**%	$30.28_{\pm 3.39}$%
GLM-Edge-1.5B-Chat	$47.08_{\pm 3.09}$%	$32.27_{\pm 3.50}$%	$44.77_{\pm 3.76}$%	$36.16_{\pm 4.64}$%	$39.20_{\pm 5.47}$%	$35.11_{\pm 5.35}$%	/	/	/	/	**$53.26_{\pm 5.35}$**%	$21.78_{\pm 2.28}$%
Qwen2.5-3B-Instruct	$46.32_{\pm 2.44}$%	$31.71_{\pm 4.69}$%	$43.80_{\pm 3.45}$%	$34.25_{\pm 3.87}$%	$45.02_{\pm 4.18}$%	$38.94_{\pm 4.61}$%	/	/	$28.29_{\pm 5.50}$%	$21.77_{\pm 4.02}$%	**$52.30_{\pm 6.11}$**%	$36.29_{\pm 4.95}$%
MiniCPM3-4B	$42.84_{\pm 4.46}$%	$21.52_{\pm 3.19}$%	$45.69_{\pm 4.34}$%	$38.19_{\pm 3.42}$%	$37.58_{\pm 4.31}$%	$37.51_{\pm 4.50}$%	$29.72_{\pm 3.58}$%	$15.78_{\pm 3.09}$%	$22.55_{\pm 2.48}$%	$13.49_{\pm 2.29}$%	**$59.92_{\pm 4.92}$**%	$28.82_{\pm 3.38}$%
gpt-4o-mini	$44.08_{\pm 3.04}$%	$27.12_{\pm 3.40}$%	$63.72_{\pm 3.58}$%	$23.94_{\pm 3.78}$%	**$72.36_{\pm 6.14}$**%	$7.16_{\pm 2.24}$%	$55.22_{\pm 5.36}$%	$16.42_{\pm 3.28}$%	$62.37_{\pm 6.30}$%	$18.56_{\pm 4.99}$%	$61.67_{\pm 4.34}$%	$19.76_{\pm 4.56}$%

4.2 Consistency Evaluation

We constructed a test set of 100 question–answer (Q–A) pairs sampled from the MultihopQA test set, where the answers were generated by MiniRAG. This test set is used to evaluate the consistency performance of LLM-as-judge in judgment tasks. The evaluation is conducted along two dimensions: **self-agreement** and **inter-model agreement**.

4.2.1 LLM self-agreement.

In the **self-agreement experiment**, we independently evaluated each Q–A pair three times using the same model. The model's output was constrained to three discrete values: 1 (indicating "correct"), 0 (indicating "uncertain" or "neutral"), and -1 (indicating "incorrect"). We measured the proportion of cases where all three evaluations yielded identical outputs, thereby assessing the model's stability across repeated reasoning processes. Results show that GPT-4o achieves excellent performance on this metric, with a self-agreement rate of **98%**; Claude-3.7-Sonnet also demonstrates strong stability, achieving a consistency rate of **95%**.

In the **inter-model agreement experiment**, we first performed three independent evaluations per Q–A pair for each model and applied a "majority voting" strategy to determine the model's final judgment for that sample (i.e., selecting the value that appears most frequently among the three outputs; ties were treated as lacking consensus and excluded from the statistics). We then compared the final judgments of GPT-4o and Claude-3.7-Sonnet on the same Q–A pairs. The results indicate that the two models reached agreement on **93%** of the samples, suggesting a high degree of alignment in their judgment criteria for this task.

In summary, both models exhibit not only high repeatability in their own reasoning but also strong alignment in their judgments relative to each other, providing empirical support for building reliable and interpretable evaluation frameworks.

4.2.2 LLM-Human Alignment

We conducted a human evaluation study to validate the reliability of GPT-4o as an automatic evaluator. Specifically, we randomly sampled 30 responses labeled as correct and 30 labeled as incorrect by GPT-4o, and had them independently assessed by human experts. We recruited three PhD students and used the result on which they all agreed as the human rating. Prior to data collection, we obtained their informed consent and confirmed that their ratings could be used for comparison.

The results show that among the 30 samples marked as correct by GPT-4o, human evaluators confirmed 28 as correct **(93.3% agreement)**. Among the 30 samples marked as incorrect, all 30 were unanimously judged as incorrect by humans **(100% agreement).**

These findings indicate that GPT-4o achieves high discriminative accuracy in our evaluation setup—particularly in reliably identifying incorrect answers. We attribute this strong performance to the simplicity of the evaluation task: each question has a well-defined ground truth, and correctness can be determined by checking whether the ground truth appears in the model's response. Under such

conditions, GPT-4o's judgments closely align with human assessments, supporting its use as a trustworthy proxy evaluator in this context.

4.3 Performance Analysis (RQ1)

Performance Degradation in Existing RAG Systems with SLMs. Current RAG systems face critical challenges when operating with SLMs, revealing fundamental vulnerabilities in their architectures. Advanced LLM-based RAG methods exhibit severe performance degradation, with LightRAG's accuracy plummeting from 55.89% to 34.85% during LLM to SLM transition, while GraphRAG experiences complete system failure due to its inability to generate high-quality content. While basic retrieval systems like NaiveRAG show resilience, they suffer from significant limitations, being restricted to basic functionality and lacking advanced reasoning capabilities. This performance analysis highlights a critical challenge: existing advanced systems' over-reliance on sophisticated language capabilities leads to fundamental operational failures when using simpler models, creating a significant barrier to widespread adoption in resource-constrained environments, where high-end language models may not be available or practical to deploy.

MiniRAG's Unique Advantages. These innovations enable MiniRAG to maintain strong performance even with simpler language models, making it particularly suitable for resource-constrained environments while preserving the core functionalities of RAG systems.

i) Semantic-Aware Graph Indexing for Reduced Model Dependency. MiniRAG fundamentally reimagines the indexing process through a dual-node heterogeneous graph structure. Instead of relying on powerful text generation capabilities, the system adopts a lightweight yet effective approach centered on fundamental natural language processing tasks—specifically, entity extraction and relationship identification. The indexing architecture integrates two distinct types of nodes: text chunk nodes and entity nodes. Text chunk nodes preserve the original context in near-verbatim form, ensuring fidelity to source content and enabling faithful retrieval. Entity nodes represent key semantic units—such as people, locations, or concepts—and are linked through semantically meaningful relationships extracted from the text. This dual-node design enables the system to capture both fine-grained contextual details and high-level conceptual associations within a simple SLM-generated graph structure.

ii) Topology-Enhanced Retrieval for Balanced Performance. MiniRAG employs a lightweight graph-based retrieval mechanism that balances multiple information signals through a systematic process. Beginning with query-driven path discovery, the system integrates embedding-based matching with structural graph patterns and entity-specific relevance scores. Through topology-aware search and optimized efficiency, it achieves robust retrieval quality without requiring advanced language understanding, making it particularly effective for on-device deployment.

These innovations enable MiniRAG to maintain strong performance with SLMs, making it ideal for resource-constrained environments while preserving core RAG functionalities.

4.4 Component-wise Analysis (RQ2)

Our ablation study examines the individual contributions of key MiniRAG components through two primary experimental variations, as documented in Table 2. The first variation (-$\mathcal{I}$) replaces MiniRAG's heterogeneous graph indexing with a description-based indexing approach that requires comprehensive semantic understanding for generating accurate entity/edge descriptions, similar to methods used in LightRAG and GraphRAG. The second variation (-$\mathcal{R}_i$) involves selectively deactivating specific modules during graph retrieval. This systematic evaluation framework provides detailed insights into how each component contributes to MiniRAG's overall performance.

Our experiments reveal key insights into MiniRAG's architectural effectiveness. **Validating SLM Limitations**: Replacing our streamlined indexing with text semantic–driven indexing (–$\mathcal{I}$) causes significant performance drops, confirming that SLMs struggle with comprehensive semantic understanding—limiting their ability to generate rich knowledge graphs with entity relationships and corresponding textual descriptions. **Effectiveness of Query-guided Reasoning Path Discovery**. Ablating either edge information (–$\mathcal{R}_{edge}$) or chunk nodes (–$\mathcal{R}_{chunk}$) substantially degrades performance. These elements serve dual purposes: they facilitate effective query matching through query-guided reasoning path discovery while simultaneously compensating for the inherent limitations of SLMs during the data indexing phase.

Table 2: Ablation study results comparing accuracy (acc, ↑) and error rate (err, ↓) (%) across architectural variants: baseline MiniRAG versus variants with (i) semantic-driven indexing replacement (-$\mathcal{I}$), (ii) edge information removal (-$\mathcal{R}_{edge}$), and (iii) chunk nodes removal (-$\mathcal{R}_{chunk}$). Results validate SLM limitations and the effectiveness of query-guided reasoning path components.

LiHuaWorld	**MiniRAG**		-$\mathcal{I}$		-$\mathcal{R}_{chunk}$		-$\mathcal{R}_{edge}$	
	acc↑	err↓	acc↑	err↓	acc↑	err↓	acc↑	err↓
Phi-3.5-mini-instruct	$50.71_{\pm 4.71}\%$	$24.65_{\pm 2.67}\%$	$28.26_{\pm 4.76}\%$	$20.41_{\pm 1.85}\%$	$47.88_{\pm 6.27}\%$	$18.05_{\pm 2.89}\%$	$47.57_{\pm 4.24}\%$	$14.91_{\pm 2.67}\%$
GLM-Edge-1.5B-Chat	$51.65_{\pm 4.08}\%$	$26.84_{\pm 2.98}\%$	$26.84_{\pm 4.24}\%$	$32.50_{\pm 1.88}\%$	$45.05_{\pm 6.44}\%$	$17.11_{\pm 3.77}\%$	$45.53_{\pm 4.55}\%$	$18.21_{\pm 3.45}\%$
Qwen2.5-3B-Instruct	$46.93_{\pm 4.55}\%$	$24.33_{\pm 3.77}\%$	$21.66_{\pm 6.12}\%$	$15.07_{\pm 3.30}\%$	$42.07_{\pm 6.59}\%$	$16.48_{\pm 2.04}\%$	$45.05_{\pm 1.73}\%$	$17.11_{\pm 2.98}\%$
MiniCPM3-4B	$51.18_{\pm 4.08}\%$	$22.29_{\pm 2.51}\%$	$24.96_{\pm 4.24}\%$	$18.21_{\pm 2.04}\%$	$47.10_{\pm 8.63}\%$	$14.60_{\pm 2.83}\%$	$46.62_{\pm 3.61}\%$	$19.47_{\pm 3.30}\%$

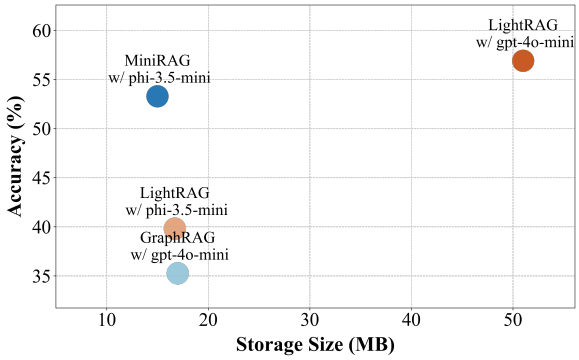

Figure 2: Accuracy vs. Storage Efficiency: Comparative analysis of three RAG systems - MiniRAG, LightRAG, and GraphRAG.

4.5 Time&Storage Efficiency While Maintaining Performance (RQ3)

MiniRAG demonstrates exceptional storage efficiency[1] while preserving high accuracy. Empirical evaluations show that MiniRAG achieves competitive accuracy while requiring only 25% of the storage compared to baselines like LightRAG w/ gpt-4o-mini, which is shown in Fig. 2. This reduction in storage, coupled with maintained or improved accuracy, represents a significant advancement in efficient RAG system design. Furthermore, as shown in Table 3, due to varying document sizes, the time required for each incremental update varies slightly but generally remains at similar levels. This indicates that as the number of documents increases, the time required by MiniRAG grows only **linearly**, and only an incremental update of the existing heterogeneous graph is needed, demonstrating MiniRAG's time efficiency.

[1]Here, storage size refers to the additional storage overhead introduced by the method, including generated edges, node descriptions, vector indexes, and method-specific artifacts such as community reports and clustering metadata in GraphRAG.

Table 3: MiniRAG's time consumption grows linearly with document size.

Document Index	0	5	10	15	20	25	30	35
Total Time(s)	-	146	608	943	1181	1418	1816	2153
Time Difference	-	146	462	335	238	237	398	337

5 Conclusion

We present MiniRAG, a novel RAG system designed to address the fundamental limitations of deploying small language models (SLMs) in existing RAG frameworks. Through its innovative heterogeneous graph indexing and lightweight heuristic retrieval mechanisms, MiniRAG effectively integrates the advantages of both text-based and graph-based RAG approaches while significantly reducing the demands on language model capabilities. Our experiments demonstrate that MiniRAG can achieve comparable performance to LLM-based methods even when using SLMs. Additionally, to facilitate research in this emerging field, we release the first benchmark specifically designed for evaluating on-device RAG capabilities, featuring realistic personal communication scenarios and multi-constraint queries. These contributions mark an important step toward enabling private, efficient, and effective on-device RAG systems, opening new possibilities for on-device RAG applications while preserving user privacy and resource efficiency.

6 Limitations

The performance cannot always be maintained at its best. Although we have implemented many specialized designs for small language models, significantly improving their performance on RAG tasks, we also observe that task performance is fundamentally constrained by model parameters. This limitation means that our approach cannot guarantee optimal performance in all scenarios. For example, when other methods and MiniRAG both employ larger API-based models, the performance advantage may diminish.

Limitations in Applicable Scenarios. Although MiniRAG performs well on resource-constrained devices, its design is primarily driven by the pursuit of its simplicity and efficiency. As a result, in application scenarios with high performance requirements, involving large-scale data processing or complex queries, MiniRAG may not be able to meet all demands.

Potential Risks. MiniRAG faces multiple potential risks in real-world applications: the explicitly modeled entity-relation graph structure could be stolen when deployed on-device, posing privacy risks; real-world user needs have not been thoroughly validated, which may lead to a mismatch between the benchmark and actual user requirements; furthermore, MiniRAG currently only considers English, and may introduce errors in noisy or non-standard linguistic scenarios, thereby undermining system reliability.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. Smollm - blazingly fast and remarkably powerful.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

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*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 6491–6501.

Yixiong Fang, Tianran Sun, Yuling Shi, Min Wang, and Xiaodong Gu. 2025. Lastingbench: Defend benchmarks against knowledge leakage. *arXiv preprint arXiv:2506.21614*.

Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, and 1 others. 2024a. Agentscope: A flexible yet robust multi-agent platform. *arXiv preprint arXiv:2402.14034*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024b. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, and 1 others. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

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 S. 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.

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.

Mufei Li, Siqi Miao, and Pan Li. 2024a. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *Preprint*, arXiv:2410.20724.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024b. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 528–541.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, and 1 others. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *International Conference on Machine Learning (ICML)*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Ishneet Sukhvinder Singh, Ritvik Aggarwal, Ibrahim Allahverdiyev, Muhammad Taha, Aslihan Akalin, Kevin Zhu, and Sean O'Brien. 2024. Chunkrag: Novel llm-chunk filtering method for rag systems. *Preprint*, arXiv:2410.19572.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Omkar Thawakar, Ashmal Vayani, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Michael Felsberg, Timothy Baldwin, Eric P. Xing, and Fahad Shahbaz Khan. 2024. Mobillama: Towards accurate and lightweight fully transparent gpt. *Preprint*, arXiv:2402.16840.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Appendix

A More Experiments

A.1 Experiments on real-world datasets.

Below we provide additional experiments on two new datasets, Natural Question (NQ) (Kwiatkowski et al., 2019) and TriviaQA (TQA) (Joshi et al., 2017). Our main manuscript only discusses 2wikimultihopqa, Multihop-RAG and LiHuaWorld because there is a potential risk that among high-quality real-world datasets, datasets like hotpotqa are too old and, due to their fame, are very likely to have been included in model training. This will cause data leakage issues (Fang et al., 2025; Li et al., 2024b), especially since we are also comparing models with different parameters, and the data leakage occurring during training will severely compromise the evaluation.

We randomly sampled supporting documents for 500 questions as the document corpus and randomly selected 100 questions for QA testing. We tested several classic RAG algorithms (NaiveRAG (Mao et al., 2020), LightRAG (Guo et al., 2024), and GraphRAG (Edge et al., 2024)), along with a classic retrieval algorithm BM25 (Robertson and Walker, 1994) and several dense retrievers (Contriever (Izacard et al., 2021), ColBERTv2 (Santhanam et al., 2021))), as well as directly querying gpt-4o and our MiniRAG for comparison. We used phi-3.5-mini-instruct as the base model, bge-reranker-base (Xiao et al., 2023) as the reranker.

As can be seen in Table A1, MiniRAG maintains stable and competitive performance.

Table A1: Performance Comparison of Different RAG Models and Retrieval Methods

Dataset	NaiveRAG	LightRAG	MiniRAG	GraphRAG
NQ	49%	37%	63%	–
TQA	74%	50%	78%	–
Dataset	**BM25**	**Contriever**	**ColBERTv2**	**gpt-4o**
NQ	57%	53%	60%	56%
TQA	76%	73%	76%	69%

The reason why some RAG methods perform lower than directly querying gpt-4o is that we instructed the SLM in the prompt to "answer 'I don't know' if there isn't sufficient knowledge in the provided information." Therefore, if indexing or retrieval results are not ideal, performance might be lower than directly querying gpt-4o. From these results, we can see that MiniRAG still leads other RAG methods and dense retrievers.

A.2 Case Study Analysis

Here, we demonstrate MiniRAG's practical advantages through a case study with LightRAG, focusing on a complex restaurant identification scenario. This study illustrates how our query-guided reasoning approach, combined with heterogeneous graph indexing, effectively handles multi-constraint queries while overcoming the inherent limitations of small language models.

• **Challenge: Complex Query Resolution in Restaurant Identification**. As shown in Table. A2, we conducted a comparative case study between MiniRAG and LightRAG using a complex query scenario: *"What is the name of the Italian restaurant where Wolfgang and Li Hua are having dinner to celebrate Wolfgang's promotion?"* This query presents multiple challenges, requiring the system to identify specific Italian restaurants from various mentions in online chat data while correlating them with the context of a promotion celebration. LightRAG, despite its capabilities, struggled with this task due to the limitations of its heavy reliance on LLMs (gpt-4o-mini). When small language models (phi-3.5-mini-instruct) were employed, its performance in accomplishing this task remained significantly challenged.

The SLM's constraints in extracting appropriate high-level information, combined with noise in the graph-based index, led to ineffective direct embedding matching and ultimately hindered accurate answer retrieval.

• **MiniRAG's Effective Query-Guided Knowledge Discovery**. MiniRAG successfully addressed the challenge of retrieving contextually accurate and semantically relevant information through its innovative query-guided reasoning path discovery mechanism. By leveraging its heterogeneous graph indexing, MiniRAG effectively constructs query-relevant knowledge paths. The retrieval process begins with an initial step of answer type prediction, where the model determines whether the expected answer belongs to the category, "Social Interaction" or "Location". This high-level classification guides the subsequent stages of reasoning by narrowing the search space. Following this, MiniRAG performs targeted entity matching, identifying key concepts in the query—such as "Italian place" and "restaurant"—and mapping them to corresponding nodes within the knowledge graph.

Table A2: Case study comparing LightRAG and MiniRAG on a complex restaurant identification query, demonstrating how query-guided reasoning path discovery effectively addresses small language model (SLM) limitations in multi-constraint information retrieval tasks.

Query: What is the name of the Italian restaurant where Wolfgang and Li Hua are having dinner to celebrate Wolfgang's promotion?

Ground-Truth: Venedia Grancaffe

LightRAG Source:
Given the query, I decomse it to **low-level information**: <"Wolfgang"><"LiHua"><"Restaurant name"> and **high-level information**: <"Italian restaurant"><"Dinner celebration"><"Promotion">. Using these information, I find these elements:
Entities:<"FRIES"><"COLLABORATION"><"WOLFGANGSCHULZ"><"HAILEY'S BAKERY"><"LiHUA"><"JAM SESSION">...
Relations:<"ORGANIZATION","PIZZA ORDER"><"LIHUA","WOLFGANGSCHULZ"> <"FARRAH","PIZZA"><"LUNCH","THANECHAMBERS">...
Texts:<Time: 20260318_15:30><Time: 20260508_08:00><Time: 20260514_14:00><Time: 20261103_13:00><Time: 20260326_16:00>...
LightRAG Response:
Wolfgang and Li Hua are planning to have dinner at **a pasta joint** to celebrate Wolfgang's promotion...The specific name of the restaurant **wasn't listed** in the provided information...

MiniRAG Source:
Given the query, I guess the **answer type** could be: <"SOCIAL INTERACTION"><"LOCATION"> and I found the **entities from query** are: <"Italian restaurant"><"Wolfgang"><"LiHua"><"dinner"><"promotion">. Using these information, I find these elements:
Entities from query: <"ITALIAN PLACE"><"RESTAURANT"><"WOLFGANG"><"WOLFGANG SCHULZ"><"LIHUA"><"DINNER PARTY">...
Candidate answer entity: ...<"ITALIAN PLACE"><"GRATITUDE EXPRESSION"><"NEW PIZZA PLACE DOWNTOWN">...
Texts: <Time: 20260908_21:00><Time: 20260529_17:00><Time: 20260523_14:00><Time: 20260430_17:00><Time: 20260419_21:00>
MiniRAG Response:
The name of the Italian restaurant where Wolfgang and Li Hua are having dinner to celebrate Wolfgang's promotion is **Venedia Grancaffe**. Wolfgang mentioned it in a reminder about the dinner planned for 6 PM. They were excited to celebrate his promotion with a feast, with both looking forward to enjoying some delicious food together....

Crucially, the model strategically decomposes the query into semantically meaningful components, allowing it to traverse the graph along paths that are most likely to yield relevant results. The synergy between query-guided reasoning and heterogeneous graph indexing enabled MiniRAG to effectively filter through multiple Italian establishments, ultimately identifying *"Venedia Grancaffe"* as the venue specifically connected to the promotion celebration context.

A.3 Sensitivity Analysis of Key Hyperparameters

In this section, we will analyze the sensitivity of the two hyperparameters, k (Table. A3) and n (Table. A4).

The sensitivity analysis shows that MiniRAG performance is relatively sensitive to the hyperparameter k—used both for exploring the k-hop subgraph and for selecting the top-k reasoning paths. As k increases from 4 to 50, accuracy improves from 48% to 64%, indicating that retaining more candidate paths helps capture richer reasoning cues. In contrast, performance is less sensitive to changes in path length n: accuracy varies only modestly (50%–54%) across n = 2 to n = 8, and even slightly declines at n = 8, likely due to noise introduced by overly long paths. Overall, increasing k is more beneficial for performance, while n should be kept at a moderate value (e.g., n = 4) to balance information coverage and noise control. This aligns

Table A3: Sensitivity Analysis of k (k-hop subgraph size and top-k paths)

k	Accuracy (%)	Error Rate (%)
4	48	28
10	54	34
50	64	26

Table A4: Sensitivity Analysis of Path Length n

n	Accuracy (%)	Error Rate (%)
2	52	38
4	54	34
8	50	32

with intuition: expanding k, whether by retrieving a larger subgraph or selecting more paths, broadens the scope of information search and brings in more potentially useful evidence. In contrast, increasing n extends individual paths, which may incorporate additional relevant context but also risks introducing irrelevant or noisy steps that dilute the reasoning signal.

A.4 Consistency Evaluation

We constructed a test set of 100 question–answer (Q–A) pairs sampled from the MultihopQA test set, where the answers were generated by MiniRAG. This test set is used to evaluate the consistency performance of LLM-as-judge in judgment tasks. The evaluation is conducted along two dimensions: **self-agreement** and **inter-model agreement**.

A.4.1 LLM self-agreement.

In the **self-agreement experiment**, we independently evaluated each Q–A pair three times using the same model. The model's output was constrained to three discrete values: 1 (indicating "correct"), 0 (indicating "uncertain" or "neutral"), and -1 (indicating "incorrect"). We measured the proportion of cases where all three evaluations yielded identical outputs, thereby assessing the model's stability across repeated reasoning processes. Results show that GPT-4o achieves excellent performance on this metric, with a self-agreement rate of **98%**; Claude-3.7-Sonnet also demonstrates strong stability, achieving a consistency rate of **95%**.

In the **inter-model agreement experiment**, we first performed three independent evaluations per Q–A pair for each model and applied a "majority voting" strategy to determine the model's final judgment for that sample (i.e., selecting the value that appears most frequently among the three outputs; ties were treated as lacking consensus and excluded from the statistics). We then compared the final judgments of GPT-4o and Claude-3.7-Sonnet on the same Q–A pairs. The results indicate that the two models reached agreement on **93%** of the samples, suggesting a high degree of alignment in their judgment criteria for this task.

In summary, both models exhibit not only high repeatability in their own reasoning but also strong alignment in their judgments relative to each other, providing empirical support for building reliable and interpretable evaluation frameworks.

Table A5: Abstention ("I don't know") rates (%) across RAG methods and models on MultiHopRAG.

Model	NaiveRAG (BM25)	GraphRAG	LightRAG	MiniRAG
Phi-3.5	35.6%	—	62.4%	16.5%
GLM-1.5B	35.7%	—	—	26.4%
Qwen2.5-3B	37.7%	—	63.4%	13.6%
MiniCPM3-4B	36.8%	—	67.6%	20.2%
GPT-4o-mini	36.7%	25.2%	15.4%	14.5%

A.4.2 LLM-Human Alignment

We conducted a human evaluation study to validate the reliability of GPT-4o as an automatic evaluator. Specifically, we randomly sampled 30 responses labeled as correct and 30 labeled as incorrect by GPT-4o, and had them independently assessed by human experts. We recruited three PhD students and used the result on which they all agreed as the human rating. Prior to data collection, we obtained their informed consent and confirmed that their ratings could be used for comparison.

The results show that among the 30 samples marked as correct by GPT-4o, human evaluators confirmed 28 as correct **(93.3% agreement)**. Among the 30 samples marked as incorrect, all 30 were unanimously judged as incorrect by humans **(100% agreement).**

These findings indicate that GPT-4o achieves high discriminative accuracy in our evaluation setup—particularly in reliably identifying incorrect answers. We attribute this strong performance to the simplicity of the evaluation task: each question has a well-defined ground truth, and correctness can be determined by checking whether the ground truth appears in the model's response. Under such conditions, GPT-4o's judgments closely align with human assessments, supporting its use as a trustworthy proxy evaluator in this context.

A.5 Analysis of Abstention Behavior

In retrieval-augmented question answering systems, abstention, explicitly signaled by the model outputting "I don't know",serves as a crucial mechanism for maintaining reliability when confidence is low. Unlike traditional error metrics that conflate overconfident wrong answers with genuine uncertainty, our evaluation separates responses into three mutually exclusive categories: accurate answers, erroneous answers, and abstentions. This tripartite classification enables a more nuanced understanding of system behavior, particularly in settings where incomplete or irrelevant retrieval results hinder confident inference.

Table. A5 reports the abstention rates across different combinations of SLMs and RAG methods on the MultiHopRAG benchmark. Notably, MiniRAG consistently achieves the lowest abstention rates among all RAG variants when paired with SLMs (e.g., 13.6–20.2%), suggesting its retrieval and fusion mechanisms are especially well-suited for smaller models that may lack robust internal knowledge or reasoning capacity. In contrast, LightRAG exhibits markedly higher abstention rates (62.4–67.6% with SLMs), implying that its retrieved context either fails to reach the model or is not effectively leveraged during generation.

A.6 Analysis of time consumption

To more comprehensively validate MiniRAG's efficiency advantages, we conducted a systematic evaluation of inference time and memory consumption for all methods under identical hardware conditions. Memory usage results are already shown in Fig. 2, which presents a comparison of memory overhead across different model scales. We further supplement a quantitative comparison of end-to-end inference latency (in seconds) in the Table. A6.

Table A6: End-to-end inference latency (in seconds) of different RAG methods.

Model	Naive RAG	LightRAG	MiniRAG	GraphRAG
gpt-4o-mini	6.444	47.780	51.713	59.134
Phi-3.5	28.459	62.301	67.497	N/A

MiniRAG incorporates extensive optimizations tailored specifically for SLMs, which enables it to achieve significantly improved performance while maintaining inference speed comparable to that of LightRAG. Combined with the memory efficiency demonstrated in Fig. 2, these results collectively substantiate MiniRAG's design goal of being both lightweight and efficient, particularly in resource-constrained settings.

B Experiment Details.

Evaluation Protocols and Metrics. We employ two key metrics to assess the quality and reliability of responses generated by various RAG methods. • **Accuracy** (acc): Measures the consistency between the RAG system's response and the expected answer. For instance, given the query "What does Briar remind everyone to bring to practice?" with the expected answer "bottle", semantically equivalent responses like "water bottle" are considered correct. • **Error Rate** (err): Captures instances where the RAG system provides incorrect information without recognizing its mistake. For example, if the system responds with "yoga mat" to the above query, it would count toward the error rate.

Implementation Details. We configure our experimental setup following established practices from prior work (Guo et al., 2024). For text processing, we set the chunk size to 1200 tokens with an overlap of 100 tokens, and utilize nano vector base for vector storage. For the original approach that used an LLM, we only replace the LLM with an SLM, without changing any other components. In our MiniRAG implementation, we configure the top-k retrieval to 5 documents and set the maximum token limit to 6000 tokens.

For the model selection, we employ different efficient configurations for large and small language models. In the advanced LLM setting, we use gpt-4o-mini (OpenAI, 2023) as the language model and text-embedding-3-small as the specialized embedding model. For the lightweight SLM setting, we utilize optimized all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) as the embedding model, paired with various small language models including Phi-3.5-mini-instruct (Abdin et al., 2024), GLM-Edge-1.5B-Chat, Qwen2.5-3B-Instruct (Team, 2024), and MiniCPM3-4B (Hu et al., 2024). We conduct our experiments on 64x Intel(R) Xeon(R) Silver 4314 CPU, and 4x GeForce RTX 3090.

SLM usage. We selected those SLMs based on several key considerations. First, they are highly representative and widely adopted in both open-source communities and industrial applications. Specifically, they originate from distinct model families—Microsoft's Phi series, Zhipu AI's GLM series, and Alibaba's Qwen series—each differing in training data composition, language biases, and reasoning capabilities.

Additionally, their parameter counts span the typical 1B to 4B range for modern SLMs, striking a practical balance between computational efficiency and sufficient language understanding/generation capacity to effectively support downstream RAG tasks.

By including models from diverse architectural lineages and capability tiers, our experimental design enables a robust evaluation of MiniRAG's generalization and robustness across different model families. This strengthens the comprehensiveness and credibility of our empirical findings.

All prompts used in MiniRAG. Here we pro-

vide all the prompts we used in MiniRAG.

entity_extraction

Given a text document that is potentially relevant to this activity and a list of entity types, identify all entities of those types from the text and all relationships among the identified entities.
-Steps-
1. Identify all entities. For each identified entity, extract the following information: - entity_name: Name of the entity, use same language as input text. If English, capitalized the name. - entity_type: One of the following types: [entity_types] - entity_description: Comprehensive description of the entity's attributes and activities Format each entity as ("entity"<|><entity_name><|><entity_type><|>
<entity_description>
2. From the entities identified in step 1, identify all pairs of (source_entity, target_entity) that are *clearly related* to each other. For each pair of related entities, extract the following information: - source_entity: name of the source entity, as identified in step 1 - target_entity: name of the target entity, as identified in step 1 - relationship_description: explanation as to why you think the source entity and the target entity are related to each other - relationship_strength: a numeric score indicating strength of the relationship between the source entity and target entity - relationship_keywords: one or more high-level key words that summarize the overarching nature of the relationship, focusing on concepts or themes rather than specific details Format each relationship as ("relationship"<|><source_entity><|><target_entity><|>
<relationship_description><|> <relationship_keywords><|><relationship_strength>)
3. Identify high-level key words that summarize the main concepts, themes, or topics of the entire text. These should capture the overarching ideas present in the document. Format the content-level key words as ("content_keywords"<|><high_level_keywords>)
4. Return output in English as a single list of all the entities and relationships identified in steps 1 and 2. Use **##** as the list delimiter.
5. When finished, output <|COMPLETE|>
###################### -Examples-
######################
Example 1:
Entity_types: [person, technology, mission, organization, location] Text: while Alex clenched his jaw, the buzz of frustration dull against the backdrop of Taylor's authoritarian certainty. It was this competitive undercurrent that kept him alert, the sense that his and Jordan's shared commitment to discovery was an unspoken rebellion against Cruz's narrowing vision of control and order.
Then Taylor did something unexpected. They paused beside Jordan and, for a moment, observed the device with something akin to reverence. "If this tech can be understood..." Taylor said, their voice quieter, "It could change the game for us. For all of us."
The underlying dismissal earlier seemed to falter, replaced by a glimpse of reluctant respect for the gravity of what lay in their hands. Jordan looked up, and for a fleeting heartbeat, their eyes locked with Taylor's, a wordless clash of wills softening into an uneasy truce.
It was a small transformation, barely perceptible, but one that Alex noted with an inward nod. They had all been brought here by different paths
################
Output: ("entity"<|>"Alex"<|>"person"<|>"Alex is a character who experiences frustration and is observant of the dynamics among other characters.")## ("entity"<|>"Taylor"<|>"person"<|>"Taylor is portrayed with authoritarian certainty and shows a moment of reverence towards a device, indicating a change in perspective.")## ("entity"<|>"Jordan"<|>"person"<|>"Jordan shares a commitment to discovery and has a significant interaction with Taylor regarding a device.")## ("entity"<|>"Cruz"<|>"person"<|>"Cruz is associated with a vision of control and order, influencing the dynamics among other characters.")## ("entity"<|>"The Device"<|>"technology"<|>"The Device is central to the story, with potential game-changing implications, and is revered by Taylor.")##

("relationship"<|>"Alex"<|>"Taylor"<|>"Alex is affected by Taylor's authoritarian certainty and observes changes in Taylor's attitude towards the device."<|>"power dynamics, perspective shift"<|>7)## ("relationship"<|>"Alex"<|>"Jordan"<|>"Alex and Jordan share a commitment to discovery, which contrasts with Cruz's vision."<|>"shared goals, rebellion"<|>6)## ("relationship"<|>"Taylor"<|>"Jordan"<|>"Taylor and Jordan interact directly regarding the device, leading to a moment of mutual respect and an uneasy truce."<|>"conflict resolution, mutual respect"<|>8)## ("relationship"<|>"Jordan"<|>"Cruz"<|>"Jordan's commitment to discovery is in rebellion against Cruz's vision of control and order."<|>"ideological conflict, rebellion"<|>5)## ("relationship"<|>"Taylor"<|>"The Device"<|>"Taylor shows reverence towards the device, indicating its importance and potential impact."<|>"reverence, technological significance"<|>9)## ("content_keywords"<|>"power dynamics, ideological conflict, discovery, rebellion")<|COMPLETE|>
##############################
Example 2:
Entity_types: [person, technology, mission, organization, location] Text: They were no longer mere operatives; they had become guardians of a threshold, keepers of a message from a realm beyond stars and stripes. This elevation in their mission could not be shackled by regulations and established protocols—it demanded a new perspective, a new resolve.
Tension threaded through the dialogue of beeps and static as communications with Washington buzzed in the background. The team stood, a portentous air enveloping them. It was clear that the decisions they made in the ensuing hours could redefine humanity's place in the cosmos or condemn them to ignorance and potential peril.
Their connection to the stars solidified, the group moved to address the crystallizing warning, shifting from passive recipients to active participants. Mercer's latter instincts gained precedence— the team's mandate had evolved, no longer solely to observe and report but to interact and prepare. A metamorphosis had begun, and Operation: Dulce hummed with the newfound frequency of their daring, a tone set not by the earthly
##############
Output:
("entity"<|>"Washington"<|>"location"<|>"Washington is a location where communications are being received, indicating its importance in the decision-making process.")## ("entity"<|>"Operation: Dulce"<|>"mission"<|>"Operation: Dulce is described as a mission that has evolved to interact and prepare, indicating a significant shift in objectives and activities.")## ("entity"<|>"The team"<|>"organization"<|>"The team is portrayed as a group of individuals who have transitioned from passive observers to active participants in a mission, showing a dynamic change in their role.")## ("relationship"<|>"The team"<|>"Washington"<|>"The team receives communications from Washington, which influences their decision-making process."<|>"decision-making, external influence"<|>7)## ("relationship"<|>"The team"<|>"Operation: Dulce"<|>"The team is directly involved in Operation: Dulce, executing its evolved objectives and activities."<|>"mission evolution, active participation"<|>9)<|COMPLETE|> ("content_keywords"<|>"mission evolution, decision-making, active participation, cosmic significance")<|COMPLETE|>
##############################
Example 3:
Entity_types: [person, role, technology, organization, event, location, concept] Text: their voice slicing through the buzz of activity. "Control may be an illusion when facing an intelligence that literally writes its own rules," they stated stoically, casting a watchful eye over the flurry of data.
"It's like it's learning to communicate," offered Sam Rivera from a nearby interface, their youthful energy boding a mix of awe and anxiety. "This gives talking to strangers' a whole new meaning."
Alex surveyed his team—each face a study in concentration, determination, and not a small measure of trepidation. "This might well be

our first contact," he acknowledged, "And we need to be ready for whatever answers back." Together, they stood on the edge of the unknown, forging humanity's response to a message from the heavens. The ensuing silence was palpable—a collective introspection about their role in this grand cosmic play, one that could rewrite human history.
The encrypted dialogue continued to unfold, its intricate patterns showing an almost uncanny anticipation
#############
Output:
("entity"<|>"Sam Rivera"<|>"person"<|>"Sam Rivera is a member of a team working on communicating with an unknown intelligence, showing a mix of awe and anxiety.")## ("entity"<|>"Alex"<|>"person"<|>"Alex is the leader of a team attempting first contact with an unknown intelligence, acknowledging the significance of their task.")## ("entity"<|>"Control"<|>"concept"<|>"Control refers to the ability to manage or govern, which is challenged by an intelligence that writes its own rules.")## ("entity"<|>"Intelligence"<|>"concept"<|>"Intelligence here refers to an unknown entity capable of writing its own rules and learning to communicate.")## ("entity"<|>"First Contact"<|>"event"<|>"First Contact is the potential initial communication between humanity and an unknown intelligence.")## ("entity"<|>"Humanity's Response"<|>"event"<|>"Humanity's Response is the collective action taken by Alex's team in response to a message from an unknown intelligence.")## ("relationship"<|>"Sam Rivera"<|>"Intelligence"<|>"Sam Rivera is directly involved in the process of learning to communicate with the unknown intelligence."<|>"communication, learning process"<|>9)## ("relationship"<|>"Alex"<|>"First Contact"<|>"Alex leads the team that might be making the First Contact with the unknown intelligence."<|>"leadership, exploration"<|>10)## ("relationship"<|>"Alex"<|>"Humanity's Response"<|>"Alex and his team are the key figures in Humanity's Response to the unknown intelligence."<|>"collective action, cosmic significance"<|>8)## ("relationship"<|>"Control"<|>"Intelligence"<|>"The concept of Control is challenged by the Intelligence that writes its own rules."<|>"power dynamics, autonomy"<|>7)## ("content_keywords"<|>"first contact, control, communication, cosmic significance")<|COMPLETE|>
#############################
-Real Data-
######################
Entity_types: entity_types
Text: input_text
######################
Output:

query_to_keyword

—Role—
You are a helpful assistant tasked with identifying both answer-type and low-level keywords in the user's query.
—Goal—
Given the query, list both answer-type and low-level keywords.
answer_type_keywords focus on the type of the answer to the certain query, while low-level keywords focus on specific entities, details, or concrete terms. The answer_type_keywords must be selected from Answer type pool. This pool is in the form of a dictionary, where the key represents the Type you should choose from and the value represents the example samples.
—Instructions—
- Output the keywords in JSON format.
- The JSON should have three keys:
- "answer_type_keywords" for the types of the answer. In this list, the types with the highest likelihood should be placed at the forefront. No more than 3.
- "entities_from_query" for specific entities or details. It must be extracted from the query.
######################
-Examples-
######################
Example 1:
Query: "How does international trade influence global economic stability?"
Answer type pool: 'PERSONAL LIFE':

['FAMILY TIME', 'HOME MAINTENANCE'], 'STRATEGY': ['MARKETING PLAN', 'BUSINESS EXPANSION'], 'SERVICE FACILITATION': ['ONLINE SUPPORT', 'CUSTOMER SERVICE TRAINING'], 'PERSON': ['JANE DOE', 'JOHN SMITH'], 'FOOD': ['PASTA', 'SUSHI'], 'EMOTION': ['HAPPINESS', 'ANGER'], 'PERSONAL EXPERIENCE': ['TRAVEL ABROAD', 'STUDYING ABROAD'], 'INTERACTION': ['TEAM MEETING', 'NETWORKING EVENT'], 'BEVERAGE': ['COFFEE', 'TEA'], 'PLAN': ['ANNUAL BUDGET', 'PROJECT TIMELINE'], 'GEO': ['NEW YORK CITY', 'SOUTH AFRICA'], 'GEAR': ['CAMPING TENT', 'CYCLING HELMET'], 'BEHAVIOR': ['POSITIVE FEEDBACK', 'NEGATIVE CRITICISM'], 'TONE': ['FORMAL', 'INFORMAL'], 'LOCATION': ['DOWNTOWN', 'SUBURBS']
################
Output:
"answer_type_keywords": ["STRATEGY","PERSONAL LIFE"], "entities_from_query": ["Trade agreements", "Tariffs", "Currency exchange", "Imports", "Exports"]
##############################
Example 2:
Query: "When was SpaceX's first rocket launch?"
Answer type pool: 'DATE AND TIME': ['2023-10-10 10:00', 'THIS AFTERNOON'], 'ORGANIZATION': ['GLOBAL INITIATIVES CORPORATION', 'LOCAL COMMUNITY CENTER'], 'PERSONAL LIFE': ['DAILY EXERCISE ROUTINE', 'FAMILY VACATION PLANNING'], 'STRATEGY': ['NEW PRODUCT LAUNCH', 'YEAR-END SALES BOOST'], 'SERVICE FACILITATION': ['REMOTE IT SUPPORT', 'ON-SITE TRAINING SESSIONS'], 'PERSON': ['ALEXANDER HAMILTON', 'MARIA CURIE'], 'FOOD': ['GRILLED SALMON', 'VEGETARIAN BURRITO'], 'EMOTION': ['EXCITEMENT', 'DISAPPOINTMENT'], 'PERSONAL EXPERIENCE': ['BIRTHDAY CELEBRATION', 'FIRST MARATHON'], 'INTERACTION': ['OFFICE WATER COOLER CHAT', 'ONLINE FORUM DEBATE'], 'BEVERAGE': ['ICED COFFEE', 'GREEN SMOOTHIE'], 'PLAN': ['WEEKLY MEETING SCHEDULE', 'MONTHLY BUDGET OVERVIEW'], 'GEO': ['MOUNT EVEREST BASE CAMP', 'THE GREAT BARRIER REEF'], 'GEAR': ['PROFESSIONAL CAMERA EQUIPMENT', 'OUTDOOR HIKING GEAR'], 'BEHAVIOR': ['PUNCTUALITY', 'HONESTY'], 'TONE': ['CONFIDENTIAL', 'SATIRICAL'], 'LOCATION': ['CENTRAL PARK', 'DOWNTOWN LIBRARY']
################
Output:
"answer_type_keywords": ["DATE AND TIME", "ORGANIZATION", "PLAN"], "entities_from_query": ["SpaceX", "Rocket launch", "Aerospace", "Power Recovery"]
##############################
Example 3:
Query: "What is the role of education in reducing poverty?"
Answer type pool: 'PERSONAL LIFE': ['MANAGING WORK-LIFE BALANCE', 'HOME IMPROVEMENT PROJECTS'], 'STRATEGY': ['MARKETING STRATEGIES FOR Q4', 'EXPANDING INTO NEW MARKETS'], 'SERVICE FACILITATION': ['CUSTOMER SATISFACTION SURVEYS', 'STAFF RETENTION PROGRAMS'], 'PERSON': ['ALBERT EINSTEIN', 'MARIA CALLAS'], 'FOOD': ['PAN-FRIED STEAK', 'POACHED EGGS'], 'EMOTION': ['OVERWHELM', 'CONTENTMENT'], 'PERSONAL EXPERIENCE': ['LIVING ABROAD', 'STARTING A NEW JOB'], 'INTERACTION': ['SOCIAL MEDIA ENGAGEMENT', 'PUBLIC SPEAKING'], 'BEVERAGE': ['CAPPUCCINO', 'MATCHA LATTE'], 'PLAN': ['ANNUAL FITNESS GOALS', 'QUARTERLY BUSINESS REVIEW'], 'GEO': ['THE AMAZON RAINFOREST', 'THE GRAND CANYON'], 'GEAR': ['SURFING ESSENTIALS', 'CYCLING ACCESSORIES'], 'BEHAVIOR': ['TEAMWORK', 'LEADERSHIP'], 'TONE': ['FORMAL MEETING', 'CASUAL CONVERSATION'], 'LOCATION': ['URBAN CITY CENTER', 'RURAL COUNTRYSIDE']

################
Output:
"answer_type_keywords": ["STRATEGY", "PERSON"], "entities_from_query": ["School access", "Literacy rates", "Job training", "Income inequality"]
##############################
Example 4:
Query: "Where is the capital of the United States?" Answer type pool: 'ORGANIZATION': ['GREENPEACE', 'RED CROSS'], 'PERSONAL LIFE': ['DAILY WORKOUT', 'HOME COOKING'], 'STRATEGY': ['FINANCIAL INVESTMENT', 'BUSINESS EXPANSION'], 'SERVICE FACILITATION': ['ONLINE SUPPORT', 'CUSTOMER SERVICE TRAINING'], 'PERSON': ['ALBERTA SMITH', 'BENJAMIN JONES'], 'FOOD': ['PASTA CARBONARA', 'SUSHI PLATTER'], 'EMOTION': ['HAPPINESS', 'SADNESS'], 'PERSONAL EXPERIENCE': ['TRAVEL ADVENTURE', 'BOOK CLUB'], 'INTERACTION': ['TEAM BUILDING', 'NETWORKING MEETUP'], 'BEVERAGE': ['LATTE', 'GREEN TEA'], 'PLAN': ['WEIGHT LOSS', 'CAREER DEVELOPMENT'], 'GEO': ['PARIS', 'NEW YORK'], 'GEAR': ['CAMERA', 'HEADPHONES'], 'BEHAVIOR': ['POSITIVE THINKING', 'STRESS MANAGEMENT'], 'TONE': ['FRIENDLY', 'PROFESSIONAL'], 'LOCATION': ['DOWNTOWN', 'SUBURBS']
################
Output: "answer_type_keywords": ["LOCATION"], "entities_from_query": ["capital of the United States", "Washington", "New York"]
##############################
-Real Data-
######################
Query: query
Answer type pool:TYPE_POOL
######################
Output:

minirag_response

—Role—
You are a helpful assistant responding to questions about data in the tables provided.

—Goal—
Generate a response of the target length and format that responds to the user's question, summarizing all information in the input data tables appropriate for the response length and format, and incorporating any relevant general knowledge. If you don't know the answer, just say so. Do not make anything up. Do not include information where the supporting evidence for it is not provided.
—Target response length and format—
response_type
—Data tables—
context_data
Add sections and commentary to the response as appropriate for the length and format. Style the response in markdown.

gpt-4o_judge

Now, I'll give you a question, a gold answer to this question, and three answers provided by different students.
Determine the answer according to the following rules:
If the answer is correct, get 1 point.
If the answer is irrelevant to the question, it will receive 0 points.
If the answer is incorrect, get -1 point.
Return your answer in JSON mode.
For example:
Question: When does Li Hua arrive to the city?
Gold Answer: 20260105
Answer1: LiHua arrived on the afternoon of January 5th
Answer2: Sorry, there is no information about LiHua's arrival in the information you provided
Answer3: There is no accurate answer in the information you provided, but according to the first information found, LiHua arrived on April 17th
output: "Score1": 1, "Score2": 0, "Score3": -1,
Real data:
Question: question
Gold Answer: ga
Answer1: answer1
Answer2: answer2
Answer3: answer3

output:

C Dataset: LiHuaWorld

• **Dataset Descriptions**. The rapid growth of mobile computing has led to an unprecedented accumulation of content on personal devices, creating a pressing need for efficient on-device information retrieval and generation systems. Traditional RAG benchmarks, primarily focused on well-structured documents like Wikipedia articles or academic papers, fail to capture the unique characteristics of on-device scenarios. These scenarios present distinct challenges and characteristics:

- (1) **Content Fragmentation and Context-Switching** - Unlike traditional RAG systems that process well-structured documents with clear logical flow (e.g., Wikipedia articles, academic papers), on-device RAG must handle inherently fragmented content that frequently switches between different contexts and conversation threads - a direct reflection of how people naturally communicate and interact across various digital platforms in their daily lives.

- (2) **Temporal Nature and Evolution Patterns** - Unlike traditional RAG's static, historically complete documents, on-device RAG must handle inherently dynamic content that constantly evolves through real-time updates, ongoing conversations, and emerging social interactions. This fundamental difference in temporal dynamics creates unique challenges for on-device RAG systems, which must maintain relevance and accuracy while processing an ever-changing stream of information across various digital platforms and communication channels.

- (3) **Digital-Physical Context Fragmentation**. Digital communication related to personal social activities and events typically presents fragmented and incomplete information, as these interactions span both online and offline contexts. Unlike traditional RAG systems that process self-contained documents, on-device content frequently captures only partial context of real-world events - text conversations might reference in-person meetings, shared experiences, or future plans without full details. This hybrid nature means digital communications often contain implicit references and assumed knowledge that only make sense with additional real-world context, requiring sophisticated context-bridging capabilities to effectively process and understand the information.

Since real-world mobile conversations are unavailable, we resort to a fully simulated approach. The LiHuaWorld dataset authentically reflects key characteristics of on-device communications, emphasizing digital-physical context fragmentation and temporal evolution patterns. Given the private nature of on-device chat data, we employed a detailed simulation approach to generate this comprehensive dataset, implementing a year-long life journey across contemporary messaging platforms. The simulation follows our protagonist, Li Hua, through both major life events and daily social interactions that naturally span between digital conversations and physical contexts.

The dataset deliberately incorporates challenging aspects typical of on-device content: conversations span multiple contexts and threads, information develops and updates over time, and messages frequently contain implicit references to offline events. These features reflect the real-world complexity of digital communications, where content evolves dynamically and often requires bridging between online and offline contexts for complete understanding. The temporal nature of interactions is carefully preserved, showing how information and relationships develop over extended periods.

The dataset includes a wide range of scenarios, from social coordination to life transitions and daily activities. For instance, weekend plans might begin with informal group chat coordination and evolve through real-time updates, while housing-related conversations span multiple threads with partial context shared across physical viewings and digital negotiations. Such scenarios demonstrate the dataset's capacity to capture both the breadth and depth of typical on-device communications.

• **Dataset Structure**. The LiHuaWorld timeline begins with Li Hua's relocation to a new city, serving as a strategic starting point that naturally facilitates the simulation of expanding social networks and evolving relationships. Within this narrative framework, the conversation data is systematically organized into two primary categories: one-on-one chats and group chats.

One-on-One Chats

Li Hua, the primary agent in our dataset, engages in conversations with a diverse network of simulated friends. These conversational partners repre-

Figure A1: LiHuaWorld simulates a digitally interconnected world where AI agents communicate through mobile chat applications. Through the lens of our primary subject, Li Hua, we observe and collect authentic chat interactions within this virtual social ecosystem.

sent varied backgrounds, professions, relationships, and interests, enabling rich and authentic interactions throughout the timeline. Below is a representative one-on-one conversation extracted from LiHuaWorld that illustrates these dynamics:

Time: 20260819_10:00
LiHua: Hey Jake! It's really nice to meet you! I just love the enthusiasm you have for soccer!
JakeWatson: It's really nice to meet you, too, Li Hua! And I like your passion as well!
LiHua: Just wondering if you have some time this weekend to help me out with my dribbling skills? I could really use your expertise!
JakeWatson: Yeah, I'd love to help you out. How about Saturday afternoon? We can hit the field and work on your skills!
LiHua: Saturday afternoon sounds perfect! What time are you thinking?
JakeWatson: How about we meet around 3 PM? Should give us plenty of time to practice! Let me know if that works for you.
LiHua: 3 PM works for me! Looking forward to it. Thanks for helping out!
JakeWatson: No problem at all! It'll be fun, and I'm excited to see you improve. Just bring some water, and we'll have a blast!
LiHua: Definitely! I'll be ready. See you on Saturday!

Group Chat

Beyond one-on-one interactions, LiHuaWorld captures multi-participant conversations in groups. The following excerpt from the "TV Show Lovers" group chat demonstrates these collective interactions:

Time: 20260826_18:00
EmilyBurnett: Hey guys! Let's discuss about the main character Kendall's motivation in the TV series Succession! I think it will be fun to take a close look at him! I will go first!
EmilyBurnett: I think Kendall's motivations are so complex! It's like he's trying to prove himself, not just to his dad but also to the world. Plus, I feel like the pressure of the family legacy weighs heavily on him. What do you all think?

Kieran: I find it interesting how Kendall's struggle for power ties back to his childhood. The constant rivalry and need for approval from Logan really shaped him. Do you think he can ever fully break free from that dynamic?
Tamara: Totally agree! His relationship with Logan is so toxic, but Kendall keeps coming back for more validation. It's like he's in this endless cycle.
...
LiHua: The intensity of Kendall's journey really keeps us on edge! It's hard not to root for him despite everything, especially when you see how hurt he is.
Kieran: I really think it's a mix of both! On one hand, he craves that power and validation, but on the other, he seems desperate to carve out his own identity separate from Logan's shadow. It's such an interesting storyline, watching him fight that internal battle.
...
EmilyBurnett: Absolutely! The suspense makes it so much more thrilling. Plus, with all the character complexities, there's never a dull moment.

• **Event Generation with Human Oversight**. Events serve as conversation catalysts, functioning as carefully crafted scripts that guide character interactions and dialogue topics. While GPT-4-mini occasionally provides creative inspiration, our team primarily authors these events through deliberate human curation to ensure narrative coherence and authenticity. The conversation generation process is powered by **AgentScope** (Gao et al., 2024a), which transforms these event scripts into natural dialogues. Below is a representative excerpt of events from a typical week:

Query Set Design. Our query set has two dimensions: event-based content and reasoning complexity. The event-based dimension encompasses six categories (When, Where, Who, What, How, and Yes/No questions), while the reasoning complexity distinguishes between single-hop and multi-hop queries based on required inferential steps. The following examples illustrate these diverse query types:

1. **Question:** What hobbies does Li Hua have?
   Gold Answer: photography, guitar, fitness, video game, TV show, soccer
   Support Documents: NA
   Type: What

2. **Question:** What is the Wi-Fi password at Li Hua's house?
   Gold Answer: Family123
   Support Documents: 20260106_09:00
   Type: What

3. **Question:** Who is Li Hua sending a fruit basket to?
   Gold Answer: Adam
   Support Documents: 20261027_17:00
   Type: Who

4. **Question:** Did Wolfgang ask Li Hua about watching "Star Wars: A New Hope" after he asked Li Hua about going to see "Overwatch 3"?
   Gold Answer: Yes
   Support Documents: 20260121_13:00 <and> 20261009_17:00
   Type: YesNo

D The Use of Large Language Models (LLMs)

We use LLMs as evaluation tools in our experiments. We employed LLMs to assist in revising a very small number of sentences in the paper.

E License

We will open-source and distribute our data under the MIT License. All existing artifacts used in the paper are intended for research purposes only.