BiqRAG: Question–Entity Bipartite Graph Retrieval-Augmented Generation for Multi-Hop Question Answering

Anonymous ACL submission

Abstract

Knowledge Graph-based Retrieval-Augmented Generation (KG-RAG) is a promising paradigm for injecting structured knowledge into Large Language Models(LLMs), especially for multi-hop Question Answering. Yet most KG-RAG methods encode knowledge as symbolic entity–relation–entity triples, causing a semantic and structural mismatch with free-form natural-language queries. This representation mismatch weakens retrieval alignment and causes errors to propagate across reasoning hops. To address this mismatch, we propose **BiqRAG**, a question–entity graph-based RAG framework that converts symbolic triples into natural-language retrieval handles. Each triple is represented as bidirectional question nodes linked to entity nodes, aligning graph retrieval with user queries while preserving entity grounding. At inference, BiqRAG decomposes complex questions, activates entity-linked candidates, and matches them within graph-constrained neighborhoods, narrowing the search space and improving multi-hop evidence discovery. Experiments on multi-hop QA datasets demonstrate that BiqRAG achieves an average improvement of 13.1 points in retrieval performance (Recall@2) and 7.93 points in question-answering performance (EM), compared with the strongest baseline.

1 INTRODUCTION

Large language models (LLMs) often struggle with knowledge-intensive tasks that require information beyond their parametric knowledge, especially when the answer depends on multiple pieces of evidence(Petroni et al., 2019; Ren et al., 2025; Li et al., 2025). Retrieval-augmented generation (RAG)(Lewis et al., 2020; Izacard et al., 2022; Borgeaud et al., 2022; Lee et al., 2025) alleviates this limitation by retrieving external knowledge at inference time and grounding generation in the retrieved evidence. Most existing RAG systems, however, rely on flat chunk-based corpora(Lewis et al., 2020), where documents are split into independent passages and retrieved primarily based on semantic similarity. While effective for single-hop fact lookup, this paradigm falls short on complex multi-hop(Zhu et al., 2025; Liu et al., 2025) question answering, where answering a query requires composing evidence and reasoning over dependencies that span multiple, often loosely connected passages.

To move beyond flat text retrieval, recent graph-based RAG methods organize external knowledge as structured graphs(Guo et al., 2024; Sun et al., 2024; Gutiérrez et al., 2024), typically by constructing knowledge graphs from $(subject, predicate, object)$(Bordes et al., 2013; Das et al., 2017) triples and retrieving evidence through entity matching and relation traversal. While such methods better capture explicit relational structure, they still rely on symbolic entity-relation representations that are not naturally aligned with natural-language retrieval queries(Berant et al., 2013; Yih et al., 2015; Wang and Han, 2025). This representation mismatch makes retrieval only weakly sensitive to the semantic intent of the query. In multi-hop reasoning, the problem is further amplified: an incorrectly retrieved intermediate fact can redirect subsequent retrieval steps, causing errors to propagate across the reasoning chain.

Consider the multi-hop query: *"When did Barcelona sign the player to whom Messi's Copa del Rey goal was compared?"* A triple-based retriever may traverse a coarse relational triple such as `(Messi's goals, compared with, Cristiano Ronaldo's goals)`, erroneously redirecting the reasoning chain toward Cristiano Ronaldo. The intended comparison, however, is finer-grained: Messi's solo goal was likened to Maradona's *"Goal of the Century"*, and the answer hinges on Barcelona's signing of Maradona in 1982.

This reveals a fundamental limitation of symbolic relations such as *"compared with"*, which collapse heterogeneous contexts into a single predicate and fail to preserve the query's fine-grained intent. In contrast, a question-level node such as *"Whose Goal of the Century was Messi's Copa del Rey solo goal compared to?"* retains the discriminative cues of the original query and steers retrieval toward the correct intermediate entity, Diego Maradona.

In this paper, we propose **BiqRAG**, a question-entity graph-based RAG framework for multi-hop question answering. The central idea is to represent corpus-derived knowledge in a form aligned with natural-language retrieval queries(Wu et al., 2025), rather than retrieving directly over symbolic triples. During indexing, BiqRAG converts each extracted triple into bidirectional natural-language question nodes and links them to the corresponding entity nodes, forming a question–entity bipartite graph in which entity nodes provide symbolic grounding while question nodes serve as semantic retrieval handles for the underlying facts. During inference, BiqRAG iteratively generates atomic sub-queries, links mentioned entities to graph nodes, activates candidate question nodes from their one-hop neighborhoods, and retrieves semantically matched question nodes as evidence for multi-hop reasoning. Our main contributions are as follows:

- We identify the **representation mismatch** between symbolic entity-relation triples and natural-language retrieval queries as a key bottleneck of existing KG-based RAG methods for multi-hop question answering, and show how it propagates errors across reasoning hops.
- To bridge this mismatch, we propose a **question–entity bipartite graph** that converts corpus-derived triples into bidirectional natural-language question nodes linked to entity nodes, aligning the retrieval space with natural-language queries while preserving symbolic grounding through entities.
- Building on this graph, we introduce an **iterative graph-grounded retrieval** mechanism that decomposes a complex query into atomic sub-queries, activates candidate question nodes from the one-hop neighborhoods of linked entities, and performs semantic matching within this structurally filtered candidate set for multi-hop reasoning.

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) enhances large language models (LLMs) by conditioning generation on external evidence obtained from a retrieval module. Existing retrievers fall into two broad categories: sparse retrievers such as TF-IDF(Sparck Jones, 1972) and BM25(Robertson and Walker, 1994), which rely on lexical term matching, and dense retrievers such as GTR(Ni et al., 2022) and Contriever(Izacard et al., 2021), which retrieve passages based on semantic similarity in a learned embedding space. While effective for single-hop queries, one-shot retrieval is often inadequate for complex multi-hop questions that require evidence from multiple sources, which has motivated a line of iterative RAG approaches that interleave retrieval with intermediate reasoning.

Several representative methods have been proposed along this direction. ReAct (Yao et al., 2022) interleaves reasoning traces with external actions, enabling the model to issue search queries dynamically based on its current reasoning state. Self-Ask (Press et al., 2023) explicitly decomposes a complex question into a sequence of follow-up subquestions that are answered step by step. IRCoT (Trivedi et al., 2023) integrates chain-of-thought reasoning with retrieval, leveraging intermediate reasoning steps to guide subsequent searches while grounding further reasoning in newly retrieved evidence. Despite their demonstrated effectiveness, these methods generally perform intermediate retrieval over unstructured, flat passage collections, rendering them susceptible to *query drift*, whereby noisy intermediate results progressively steer subsequent retrieval steps away from the original information need.

2.2 Graph-structured RAG

Recent work has introduced graph-based RAG methods(Han et al., 2024; Gutiérrez et al., 2025; Ma et al., 2025) into RAG to better support complex reasoning. LightRAG(Guo et al., 2024) constructs a lightweight graph index and retrieves evidence through entity-level and relation-level matching. HippoRAG(Gutiérrez et al., 2024) represents long-term memory as a knowledge graph and performs graph-based evidence retrieval through spreading activation or personalized PageRank(Jeh and Widom, 2003). KG-RAG(Sanmartin, 2024) methods link query en-

tities to knowledge graphs and expand neighboring triples or relation paths, while Think-on-Graph(Sun et al., 2024) lets LLMs iteratively traverse relations and entities for multi-hop reasoning. GNN-RAG(Mavromatis and Karypis, 2024) and G-Retriever-style(Xu et al., 2024) methods retrieve relevant subgraphs and use graph neural networks or LLMs to reason over graph-grounded evidence. These approaches make relational dependencies more explicit than flat retrieval, but they typically operate over symbolic graph units such as entities, relations, triples, paths, or subgraphs. Such representations are not naturally aligned with natural-language queries, which can weaken semantic matching and amplify errors across reasoning hops. BiqRAG addresses this limitation by transforming triples into bidirectional natural-language question nodes while retaining entity nodes for symbolic grounding.

3 Method

In this section, we introduce BiqRAG, a question-centric graph-based RAG framework that organizes corpus-derived knowledge as a question–entity bipartite graph for multi-hop question answering. An overview of the framework is illustrated in Figure 1.

3.1 Indexing

Triple Extraction. Given a corpus segmented into chunks $\mathcal{C} = \{c_i\}_{i=1}^{N}$, we perform OpenIE(Etzioni et al., 2008; Angeli et al., 2015)-style triple extraction on each chunk using an LLM-based extractor. Specifically, the LLM is prompted to identify relational facts expressed in the chunk and output them in a structured triple format. The same LLM backbone as specified in Section 4.1 is used for this extraction step, and the extraction prompt is provided in Appendix A4. Each extracted fact is represented as:

$$\tau = (h, r, t, c) \tag{1}$$

where h and t denote the head entity and tail entity, r denotes the relation phrase, and c is the source chunk from which the triple is extracted. We keep the source chunk as provenance for the triple. This provenance is later attached to the generated question nodes and used as supporting evidence during inference.

Bidirectional Question Generation. A raw triple (h, r, t) only provides a compact symbolic representation of a fact, but it does not naturally match the semantic form of user questions. To bridge this gap, we convert each triple into two complementary natural-language question nodes. Specifically, for each triple $\tau = (h, r, t, c)$, we prompt an LLM to generate:

- $q_{h\to t}$: a question that takes h as the known anchor entity and whose answer is t;
- $q_{t\to h}$: a question that takes t as the known anchor entity and whose answer is h.

Generating questions in both directions enables the underlying fact to support reasoning from either entity. This bidirectional formulation is crucial, as the reasoning trajectory of a multi-hop question is typically unknown a **priori**. Furthermore, by conditioning the generation on both the extracted triple and its source chunk c, the resulting questions preserve fine-grained contextual cues—such as event context, entity disambiguation, and relation-specific phrasing—that are often discarded in pure symbolic representations.

Bipartite Graph Construction. Conceptually, a bipartite graph partitions its nodes into two disjoint subsets such that edges only exist between nodes of different subsets. Formally, an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as bipartite if its node set $\mathcal{V}$ can be partitioned into two subsets $\mathcal{U}$ and $\mathcal{W}$ satisfying:

$$\mathcal{V} = \mathcal{U} \cup \mathcal{W} \quad \mathcal{U} \cap \mathcal{W} = \emptyset \tag{2}$$

and $\mathcal{E} \subseteq \{(u, w) \mid u \in \mathcal{U}, w \in \mathcal{W}\}$

Equivalently, no edges exist between nodes within the same subset. Building upon this definition, BiqRAG instantiates this structure as an undirected bipartite graph comprising two distinct node types: entity nodes and question nodes. In this design, entities serve as symbolic grounding points, whereas generated questions act as semantic retrieval handles. By enforcing this bipartite separation, the graph inherently precludes direct entity–entity or question–question connections. Consequently, retrieval originating from an entity is strictly constrained to its neighboring question nodes, allowing the graph topology to function as an effective first-stage filter prior to fine-grained semantic matching. Formally, we construct:

$$\mathcal{G} = \big(\mathcal{V}_E \cup \mathcal{V}_Q, \mathcal{E}_{\text{anc}} \cup \mathcal{E}_{\text{ans}}\big) \tag{3}$$

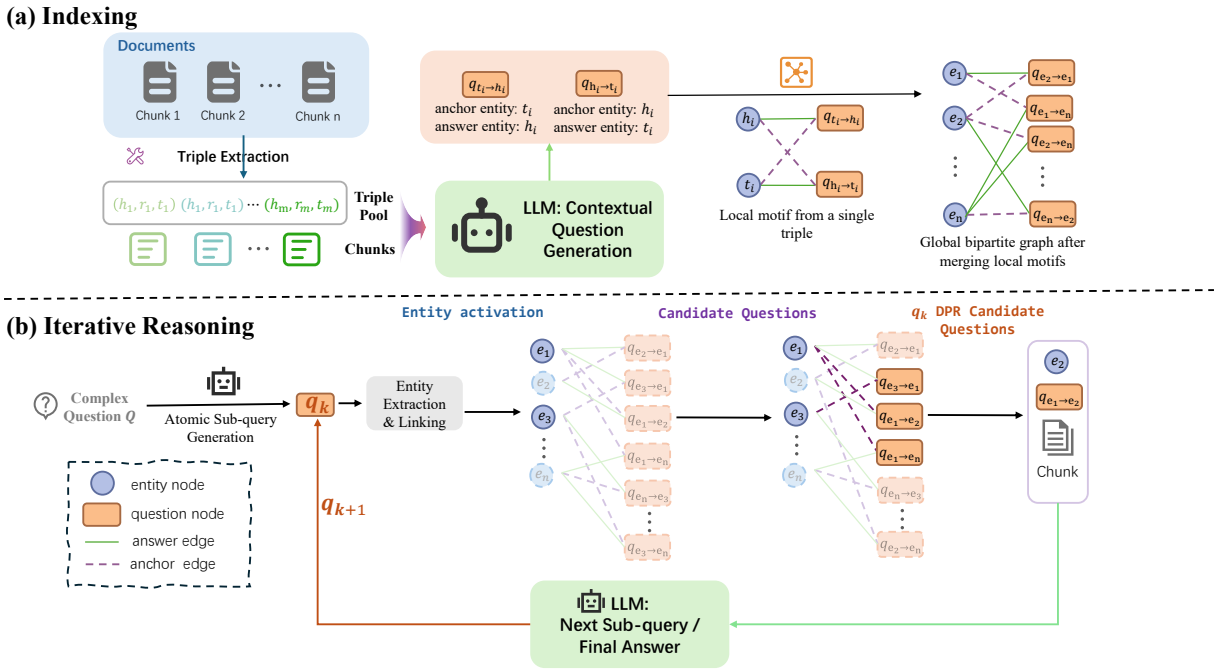

Figure 1: The Workflow of BiqRAG. BiqRAG builds a question–entity bipartite graph by converting corpus-derived triples into bidirectional question nodes. During reasoning, entities activate neighboring question nodes for semantic retrieval, and the matched nodes return source chunks as evidence.

where $\mathcal{V}_E$ is the set of entity nodes, $\mathcal{V}_Q$ is the set of generated question nodes, and $\mathcal{V}_E \cap \mathcal{V}_Q = \emptyset$. The graph is undirected, and both edge sets consist of unordered entity–question pairs:

$$\mathcal{E}_{\text{anc}} \subseteq \{\{e, q\} \mid e \in \mathcal{V}_E, q \in \mathcal{V}_Q\} \quad (4)$$

$$\mathcal{E}_{\text{ans}} \subseteq \{\{e, q\} \mid e \in \mathcal{V}_E, q \in \mathcal{V}_Q\} \quad (5)$$

Although the graph is structurally undirected, its edges are typed to encode distinct semantic roles. Specifically, **anchor edges** link a question to the entity serving as its known clue, whereas **answer edges** connect the question to the entity that constitutes its answer.

For each triple (h, r, t, c) and its generated question nodes $q_{h\to t}$ and $q_{t\to h}$, we add the following typed undirected edges:

$$\{h, q_{h\to t}\} \in \mathcal{E}_{\text{anc}} \quad \{t, q_{h\to t}\} \in \mathcal{E}_{\text{ans}} \quad (6)$$

$$\{t, q_{t\to h}\} \in \mathcal{E}_{\text{anc}} \quad \{h, q_{t\to h}\} \in \mathcal{E}_{\text{ans}} \quad (7)$$

Consequently, the graph explicitly separates the entity that conditions a question from the entity that answers it, while maintaining an undirected topology. During inference, a given entity retrieves its one-hop neighboring question nodes via anchor edges, and semantic matching is performed exclusively within this structurally filtered candidate set. Once a question node is successfully matched, its corresponding answer edge identifies the candidate answer entity.

Furthermore, because each question node is associated with its source chunk c, these nodes serve as the primary targets for semantic retrieval, while their underlying chunks provide the textual evidence for final answer generation.

3.2 Iterative Reasoning on the Question–Entity Bipartite Graph

Following the constructing the question–entity bipartite graph, BiqRAG performs multi-hop QA through iterative graph-grounded retrieval. At each step, a sub-query activates relevant entities to retrieve matching question nodes as evidence, until the answer can be produced or the maximum step limit is reache

Atomic Sub-query Generation. Given a complex question Q, BiqRAG decomposes the overall reasoning process into a sequence of atomic sub-queries. We define an atomic sub-query as a minimal, fact-seeking query corresponding to a single reasoning hop on the question–entity bipartite graph; that is, it can be resolved by matching a local question node adjacent to an activated entity. At reasoning step k, BiqRAG prompts the

Method	MuSiQue		2Wiki		HotpotQA		Average	
	Recall@2	Recall@5	Recall@2	Recall@5	Recall@2	Recall@5	Recall@2	Recall@5
BM25	33.6	43.5	53.1	65.2	56.8	74.8	47.8	61.2
Contriever	34.8	46.6	46.6	57.5	58.1	75.3	46.5	59.8
GTR	37.4	49.1	60.2	67.9	59.6	73.9	52.4	63.6
IRCOT	40.5	54.7	53.0	67.2	65.6	81.7	53.0	67.9
HippoRAG	41.0	52.1	71.5	89.5	60.4	77.3	57.6	73.0
IRCOT+HippoRAG	43.9	56.6	75.3	93.4	64.7	81.5	61.3	77.2
BiqRAG	**56.8**	**66.6**	**83.3**	**93.9**	**83.0**	**84.7**	**74.4**	**81.7**

Table 1: We report the retrieval performance metrics Recall@2 and Recall@5. For methods requiring language model integration (specifically, the bottom four approaches in the table), we employ GPT-3.5-Turbo-1106. The highest scores are bolded, while the second-highest are underlined.

LLM to generate the current atomic sub-query, denoted as $q_k = \mathrm{LLM}(Q, \mathcal{H}_{k-1})$, where $\mathcal{H}_{k-1}$ denotes the accumulated reasoning history, including previously generated sub-queries, retrieved chunks, and question–answer information.

Entity Extraction and Linking. Given the atomic sub-query q_k, BiqRAG extracts entity mentions using a named entity recognition (NER) module, denoted as $\mathcal{M}_k = \mathrm{NER}(q_k)$, where $\mathcal{M}_k$ is the set of entity mentions detected in q_k. The concrete implementation of the NER system follows the LLM configuration described in Setting 4.1.

BiqRAG then links these mentions to entity nodes in the question–entity bipartite graph. Since entity mentions may have aliases, synonyms, or different surface forms, a mention can correspond to multiple possible entity nodes. Therefore, for each extracted mention, we retain the top-K_E linked entity nodes instead of relying on a single exact match.

For each mention $m \in \mathcal{M}_k$, BiqRAG computes a linking score with each entity node $e \in \mathcal{V}_E$:

$$\phi(m, e) = \mathrm{sim}(g(m), g(e)) \tag{8}$$

where $g(\cdot)$ denotes the entity representation function. In our implementation, $g(\cdot)$ is instantiated with the Contriever model, which encodes each entity into a dense semantic embedding vector. The activated entity set at reasoning step k is defined as the union of the top-K_E linked entities for all extracted mentions:

$$\mathcal{A}_k = \bigcup_{m \in \mathcal{M}_k} \mathrm{TopK}_{e \in \mathcal{V}_E}^{K_E} \phi(m, e) \tag{9}$$

where $\mathcal{A}_k \subseteq \mathcal{V}_E$ denotes the set of activated entity nodes. These activated entities serve as symbolic grounding points for the current reasoning step.

Candidate Question Activation. Given the activated entity set $\mathcal{A}_k$, BiqRAG collects candidate question nodes through anchor edges. Specifically, the candidate set at step k is defined as:

$$\mathcal{C}_k = \{q \in \mathcal{V}_Q \mid \exists e \in \mathcal{A}_k, \{e, q\} \in \mathcal{E}_{\text{anc}}\} \tag{10}$$

Thus, only one-hop question nodes anchored by the activated entities are considered for subsequent semantic matching.

Semantic Matching. BiqRAG performs semantic matching between the sub-query q_k and the candidate question nodes in $\mathcal{C}_k$, and selects the best-matched question node q_k^*, whose associated source chunk $c(q_k^*)$ is taken as the retrieved evidence at step k.

Evidence Aggregation. For the matched question node q_k^*, BiqRAG retrieves its associated source chunk $c(q_k^*)$ stored as provenance during indexing. This chunk, together with the matched question and its answer entity (identified via the answer edges E_{ans}), is appended to the reasoning history:

$$\mathcal{H}_k = \mathcal{H}_{k-1} \cup \{\, q_k,\, q_k^*,\, c(q_k^*) \,\} \tag{11}$$

which serves as the context for the next reasoning step.

Answer Generation. At each step, the LLM inspects the updated reasoning history $\mathcal{H}_k$. If it judges that the accumulated evidence is sufficient to answer the original question Q, it stops generating new sub-queries and directly produces the final answer $\hat{a} = \mathrm{LLM}(Q, \mathcal{H}_k)$; otherwise, it generates the next sub-query q_{k+1} and the iterative retrieval continues.

Method	MuSiQue		2Wiki		HotpotQA		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
IRCOT	15.45	24.09	26.50	35.51	46.52	61.21	29.49	40.27
HippoRAG	19.40	30.12	45.50	58.65	42.00	56.51	35.63	48.42
HippoRAG+IRCOT	22.30	33.68	48.30	64.35	46.60	61.09	39.07	53.04
BiqRAG	**34.00**	**47.87**	**48.50**	**65.52**	**58.50**	**72.06**	**47.00**	**61.82**

Table 2: We report the QA performance metrics EM and F1 score with GPT-3.5-Turbo-1106 and top 5 passages here, where the best score is in bold and the second best is underlined.

4 Experiments

4.1 Experimental Setup

Datasets We collect three widely used multi-hop question answering datasets: HotpotQA(Yang et al., 2018), 2WikiMultiHopQA(Ho et al., 2020), and MuSiQue(Trivedi et al., 2022). These datasets require models to retrieve and combine multiple pieces of evidence across different passages, making them suitable benchmarks for evaluating graph-based retrieval and multi-hop reasoning ability. Following prior work, we sample 1,000 questions from the validation set of each dataset for evaluation. All subsequent experiments are conducted on these sampled subsets.

Baseline We compare BiqRAG with six representative baselines. BM25(Robertson and Walker, 1994) is a sparse lexical retriever based on term matching. Contriever(Izacard et al., 2021) and GTR(Ni et al., 2022) are dense retrievers that retrieve passages based on semantic similarity in a learned embedding space. IRCoT(Trivedi et al., 2023) is an iterative retrieval method that alternates between chain-of-thought reasoning and retrieval to collect multi-hop evidence. HippoRAG(Gutiérrez et al., 2024) is a graph-based retrieval method that builds a knowledge graph from the corpus and retrieves evidence through graph-based activation over query-related entities. IRCoT+HippoRAG(Gutiérrez et al., 2024) further combines iterative reasoning with graph-based retrieval.

Metrics We report retrieval and QA performance on the datasets above using Recall@2 and Recall@5 (R@2 and R@5 below) for retrieval and exact match (EM) and F1 scores for QA performance.

Settings For a fair comparison, we adopt Contriever as the default retriever for IRCoT, HippoRAG, and BiqRAG. For all methods involving LLM calls, we keep the LLM backbone fixed throughout the entire pipeline within each experimental setting. Specifically, once an LLM backbone is selected, the same model is consistently employed across all LLM-dependent components, including named entity extraction and linking, triple extraction, graph construction, intermediate reasoning, and final answer generation. This consistency applies to IRCoT, HippoRAG, and BiqRAG. At inference time, we set top-$K_E = 5$ for entity extraction and linking in BiqRAG, and each method retrieves the top-5 documents as supporting evidence, from which the final answer is generated. In the main experiments, we use GPT-3.5-Turbo (Ye et al., 2023; Brown et al., 2020) as the LLM backbone for the full pipeline. In Appendix A2, we further report the retrieval performance of IRCoT and BiqRAG when the entire pipeline is instantiated with GPT-4o-mini (OpenAI et al., 2024).

4.2 Main Results

Retrieval Performance Table 1 reports retrieval performance in terms of Recall@2 and Recall@5 on three multi-hop QA benchmarks: MuSiQue, 2Wiki, and HotpotQA. BiqRAG consistently outperforms all baselines across every dataset and metric, achieving 74.4 average Recall@2 and 81.7 average Recall@5, surpassing the strongest baseline (IRCoT+HippoRAG) by 13.1 and 4.5 absolute points, respectively. The gains are most pronounced on MuSiQue, the most challenging benchmark, with improvements of 12.9 and 10.0 points on Recall@2 and Recall@5, and on HotpotQA, where BiqRAG yields a striking 17.4-point Recall@2 gain over the strongest baseline (83.0 vs. 65.6).

Among baselines, dense retrievers exhibit mixed performance: Contriever is comparable to or slightly weaker than BM25 on average, while GTR provides moderate improvements. Iterative retrieval with IRCoT brings additional gains, partic-

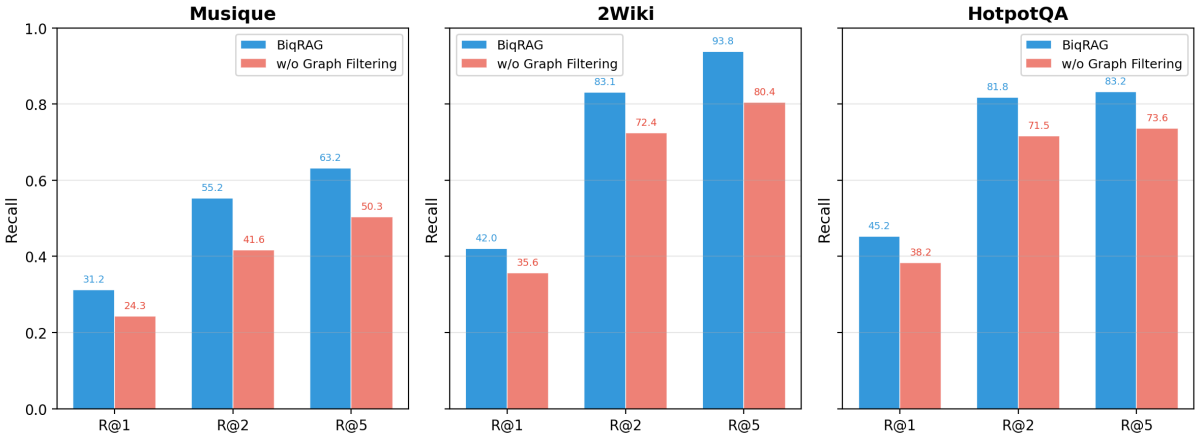

Figure 2: Comparison of BiqRAG and a variant without graph filtering (w/o GF) on MuSiQue, 2Wiki, and HotpotQA, evaluated with GPT-4o-mini.

ularly on HotpotQA, but remains substantially behind graph-augmented methods. Graph-augmented approaches (HippoRAG, IRCoT+HippoRAG) perform stronger overall, confirming the value of structured knowledge for multi-hop retrieval. BiqRAG further surpasses both, suggesting that explicitly modeling question–entity relations is more effective than relying solely on entity-level triples.

Early-rank retrieval precision. The gap between Recall@2 and Recall@5 reveals a further advantage of BiqRAG. While baselines gain substantially when expanding from top-2 to top-5 (e.g., IRCoT+HippoRAG improves by 15.9 points on average), BiqRAG gains only 7.3 points, indicating that the most relevant evidence is already concentrated at the top ranks. This shows that BiqRAG not only retrieves more relevant documents but ranks them more precisely—a property especially valuable for downstream generation, where LLMs are sensitive to context ordering and conciseness.

QA performance As shown in Table 2, BiqRAG achieves the strongest downstream QA performance, attaining an average of 47.00 EM and 61.82 F1. These results correspond to substantial improvements of 7.93 EM and 8.78 F1 over the best-performing baseline, HippoRAG+IRCoT. We attribute this performance gain primarily to BiqRAG's superior early-rank retrieval precision. Since the final answer is generated from the top-5 retrieved passages, simply retrieving the correct evidence is not sufficient—the evidence must also be ranked sufficiently high to be effectively leveraged by the LLM. Notably, BiqRAG attains the highest Recall@2 across all datasets, indicating that gold evidence is more consistently placed at the top of the input context. This ranking advantage is particularly important, as LLMs are known to be highly sensitive to context ordering and tend to underutilize relevant information when it appears in later positions or is interleaved with noisy passages. Consequently, BiqRAG's question–entity bipartite retrieval not only improves evidence coverage but also enhances the accessibility of evidence during generation, ultimately leading to higher EM and F1 scores.

5 Ablation Studies

We conduct ablation studies on the three multi-hop datasets (MuSiQue, 2Wiki, and HotpotQA) to validate our key design choices, with results reported in Figure 2 and Figure 3. Unless otherwise specified, all ablation experiments are conducted with GPT-4o-mini as the LLM backbone to better isolate the effect of individual design choices under a stronger reasoning model. As a result, the absolute numbers in this section are not directly comparable to those reported in Table 1, which use GPT-3.5-turbo.

Effect of Graph Filtering To assess the importance of graph-structured candidate filtering, we ablate this mechanism and allow semantic retrieval to operate over the entire question-node pool rather than the one-hop neighborhoods of linked entities. We denote this variant as **w/o GF**.

As shown in Figure 2, removing graph filtering consistently degrades retrieval performance across all three datasets, with the most pronounced drop on MuSiQue. This confirms that the bipartite graph structure is pivotal in constraining the retrieval space during iterative reasoning: without it, semantically similar yet structurally irrelevant nodes are

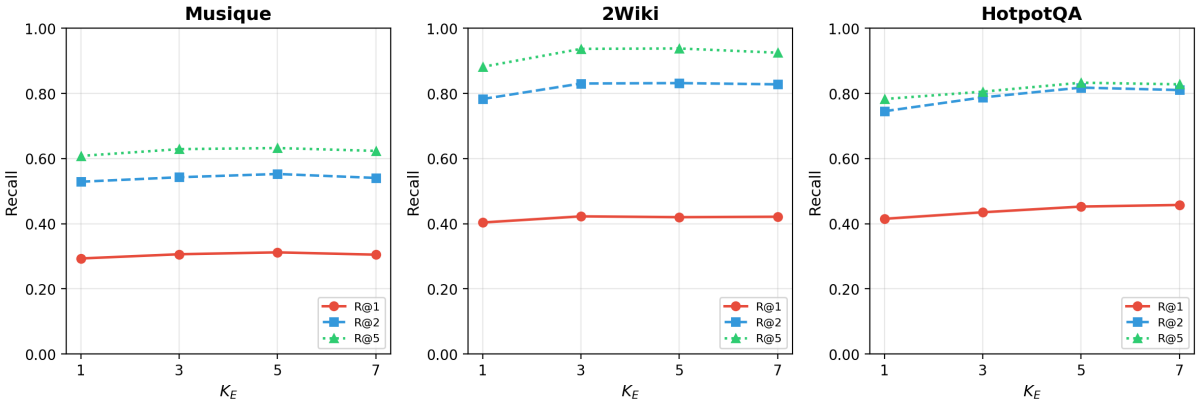

Figure 3: Ablation study on the entity linking breadth K_E across MuSiQue, 2Wiki, and HotpotQA, evaluated with GPT-4o-mini.

more likely to be retrieved, amplifying semantic drift across hops.

In contrast, our graph-filtered retrieval anchors each sub-query to its linked entities and activates only adjacent question nodes as candidates, allowing dense retrieval to focus on locally pertinent evidence. The gain is especially salient on MuSiQue, where deeper compositional reasoning makes intermediate retrieval errors more costly. These results indicate that BiqRAG's improvements arise not from semantic retrieval alone, but from its synergy with graph-grounded structural filtering.

Effect of K_E. Figure 3 investigates the impact of the entity-linking breadth K_E, which governs the number of entity nodes retained for candidate question activation at each reasoning step. We sweep $K_E \in \{1, 3, 5, 7\}$ on all three datasets and report Recall@2 and Recall@5 under the GPT-4o-mini backbone.

Overall trend. Across all three datasets, retrieval performance follows a clear *rise-then-fall* pattern as K_E increases, with the peak consistently occurring at $K_E = 5$. This reveals an intrinsic trade-off between *entity coverage* (small K_E misses gold entities) and *candidate-set noise* (large K_E pulls in irrelevant neighborhoods).

We take HotpotQA as a representative case for quantitative analysis. When $K_E = 1$, Recall@2 drops sharply by **7.3** points (0.745 vs. 0.818) and Recall@5 by **5.0** points (0.783 vs. 0.833) relative to the best configuration. This is because entity mentions in atomic sub-queries often exhibit surface-form variation (aliases, abbreviations, partial names); a restrictive K_E forces the linker to commit to a single candidate, and once the correct entity is missed, the corresponding one-hop question neighborhood is never activated, breaking the reasoning chain at its first hop. Conversely, enlarging K_E from 5 to 7 reduces Recall@2 by **0.8** points (0.810 vs. 0.818) and Recall@5 by 0.5 points (0.828 vs. 0.833), since low-confidence entity candidates pull in structurally irrelevant neighborhoods and inflate the candidate question set with semantically similar but contextually wrong nodes, weakening the filtering effect of the bipartite graph. We therefore adopt $K_E = 5$ as the default, which balances lexical coverage and candidate-set purity.

6 Conclusion

In this paper, we presented BiqRAG, a question–entity bipartite graph-based RAG framework that addresses the representation mismatch between symbolic knowledge-graph triples and natural-language queries in multi-hop QA. By converting corpus-derived triples into question nodes linked to entity nodes, BiqRAG aligns the retrieval space with user queries while preserving symbolic grounding. On this graph, we further introduced an iterative retrieval mechanism that decomposes complex questions into atomic sub-queries and performs semantic matching over entity-grounded one-hop neighborhoods, effectively mitigating semantic drift across reasoning hops. Our findings suggest that bridging the symbolic–semantic gap through question-centric graph representations is a promising direction for advancing multi-hop reasoning in retrieval-augmented generation, and we hope BiqRAG inspires future work on integrating graph-based knowledge organization with natural-language-aligned retrieval for more robust and interpretable LLM reasoning.

7 Limitations

Early convergence in iterative reasoning. BiqRAG concentrates relevant evidence in the top ranks, achieving the strongest Recall@2 (74.4) among all methods. However, the marginal gain from Recall@2 to Recall@5 is only $+7.3$ points, substantially smaller than that of the strongest baselines (e.g., $+15.4$ for HippoRAG and $+15.9$ for IRCOT+HippoRAG). This indicates that when the correct evidence is not retrieved within the first one or two hops, subsequent iterations often fail to recover: inaccurate intermediate facts entering the reasoning history H_{k-1} lead the LLM to generate semantically redundant sub-queries, forming a **semantic dead loop**, while one-hop-constrained retrieval offers no mechanism to escape an incorrect subgraph.

Indexing cost of bidirectional question generation. Generating two question nodes per triple roughly doubles the offline indexing cost compared with purely symbolic graph construction, which may hinder scalability to very large or frequently updated corpora.

Reliance on LLMs throughout the pipeline. BiqRAG uses LLMs for triple extraction, triple extraction, question generation, sub-query decomposition, and answer generation, making overall performance sensitive to the backbone LLM's capability.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3).

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569.

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*.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, and 1 others. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

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*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and

Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

Glen Jeh and Jennifer Widom. 2003. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279.

Zhicheng Lee, Shulin Cao, Jinxin Liu, Jiajie Zhang, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. 2025. Rearag: Knowledge-guided reasoning enhances factuality of large reasoning models with iterative retrieval augmented generation. *arXiv preprint arXiv:2503.21729*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See Kiong Ng, Tat-Seng Chua, and Yang Deng. 2025. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5131–5157.

Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pages 52782–52806.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and 1 others. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2463–2473.

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

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2025. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3697–3715.

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.

Diego Sanmartin. 2024. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, volume 2024, pages 3868–3898.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.

Jingjin Wang and Jiawei Han. 2025. Proprag: Guiding retrieval with beam search over proposition paths. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6223–6238.

Yaxiong Wu, Jianyuan Bo, Yongyue Zhang, Sheng Liang, and Yong Liu. 2025. Query-centric

graph retrieval augmented generation. *Preprint*, arXiv:2509.21237.

Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 2905–2909.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

S Yao, J Zhao, D Yu, N Du, I Shafran, K Narasimhan, and YR Cao. 2022. Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, and 1 others. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924.

Appendix

Appendix Contents

- A1 Datasets
- A2 GPT-4o-mini Retrieval Results
- A3 Retrieval Efficiency
- A4 Prompts

A1 Datasets

Datasets	Musique	2Wiki	hotpotqa
number	1000	1000	1000
docs	11,656	6,119	9,811
nodes-q	253,214	126,570	244,692
nodes-e	103,104	50,322	97,348
Avg. anchor edges per node-e	2.46	2.51	2.51

Table 3: Dataset Statistics.

Table 3 reports the statistics of the indexed graphs on the three datasets. Although BiqRAG generates a large number of question nodes, ranging from 126K to 253K, these nodes are grounded by entity nodes through anchor edges. The average number of anchor-connected question nodes per entity is only about 2.5 across all datasets. Therefore, during inference, once an entity is linked, BiqRAG only needs to perform semantic matching within a very small one-hop neighborhood rather than over the entire question-node pool. This shows that the question–entity bipartite graph provides an effective structural filter, substantially reducing the candidate space before dense retrieval.

A2 GPT-4o-mini Retrieval Results

Method	Recall@1	Recall@2	Recall@5
IRCoT	27.9	43.0	56.7
BiqRAG	31.2	55.3	63.2

Table 4: Retrieval performance with GPT-4o-mini on Musique.

Method	Recall@1	Recall@2	Recall@5
IRCoT	35.9	53.5	66.8
BiqRAG	42.0	83.1	93.8

Table 5: Retrieval performance with GPT-4o-mini on 2Wiki.

Tables 4-6 report the retrieval results under GPT-4o-mini. BiqRAG consistently outperforms IRCoT on Recall@2 across all three datasets, with especially large gains on MuSiQue and 2Wiki. This indicates that the advantage of BiqRAG is not tied to a specific LLM backbone, but comes from the question–entity bipartite graph and its graph-filtered retrieval mechanism. On HotpotQA, although IRCoT obtains a slightly higher Recall@5, BiqRAG still achieves better Recall@1 and Recall@2, showing that it ranks relevant evidence earlier in the retrieved list.

Method	Recall@1	Recall@2	Recall@5
IRCoT	43.4	72.8	86.2
BiqRAG	45.3	81.8	83.3

Table 6: Retrieval performance with GPT-4o-mini on hotpotqa.

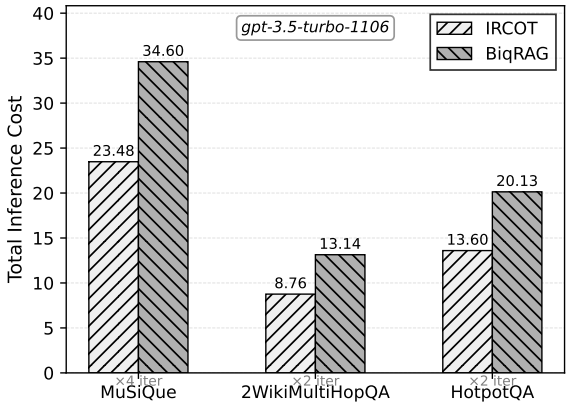

Figure 4: Total inference cost comparison between IRCoT and BiqRAG using GPT-3.5-Turbo-1106 (costs reported in USD). The maximum number of reasoning iterations for each dataset is shown above the corresponding bars.

A3 Retrieval Efficiency

Figure 4 compares the total online inference cost of IRCoT and BiqRAG under the same LLM backbone, gpt-3.5-turbo-1106. We use the same maximum number of reasoning iterations for both methods, with 4 iterations on MuSiQue and 2 iterations on 2WikiMultiHopQA and HotpotQA.

Compared with IRCoT, BiqRAG introduces additional graph-grounded retrieval operations, including entity linking, candidate question activation, and semantic matching over question nodes. As a result, BiqRAG incurs a moderate increase in total inference cost: from 23.48 to 34.60 on MuSiQue, from 8.76 to 13.14 on 2WikiMultiHopQA, and from 13.60 to 20.13 on HotpotQA. Overall, the cost is about 1.5 times that of IRCoT.

However, this additional cost is bounded by the graph filtering mechanism. Instead of matching against the entire question-node pool, BiqRAG only performs semantic retrieval within the one-hop neighborhoods of linked entities. As shown in Table 3, each entity is connected to only about 2.5 anchor question nodes on average, which greatly reduces the candidate space during retrieval. Therefore, BiqRAG trades a moderate inference-cost increase for substantially better retrieval quality, especially in early-rank recall.

A4 Prompts

Listings 1-4 report all prompts used in BiqRAG. Specifically, Listing 1 shows the prompt for named entity recognition, Listing 2 presents the prompt for corpus-level triple extraction, Listing 3 gives the prompt for generating bidirectional atomic question nodes from extracted triples, and Listing 4 provides the prompt used during iterative reasoning and sub-query generation. These prompts are kept fixed across all datasets and experimental settings. For components that require structured outputs, such as entity extraction, triple extraction, and question-node generation, we explicitly require JSON-formatted responses to facilitate automatic parsing and graph construction. During inference, the reasoning prompt constrains the model to generate one atomic sub-query at each step or terminate with a final answer when sufficient evidence has been collected.

Listing 1: Prompt for NER

```
Your task is to extract named entities from the
given paragraph.
Respond with a JSON list of entities.
Case Learning:
Inputs:
Radio City
Radio City is India's first private FM radio
station and was started on 3 July 2001.
It plays Hindi, English and regional songs.
Radio City recently forayed into New Media in
May 2008 with the launch of a music portal -
PlanetRadiocity.com that offers music related
news, videos, songs, and other music-related
features.
Outputs:
{"named_entities":
    ["Radio City","India","3 July 2001","Hindi
","English","May 2008", "PlanetRadiocity.com"]
}
```

Listing 2: Prompt for Triple Extraction

```
Your task is to construct an RDF (Resource
Description Framework) graph from the given
passages and named entity lists.
Respond with a JSON list of triples, with each
triple representing a relationship in the RDF
graph.
Pay attention to the following requirements:
1:Each triple should contain at least one, but
preferably two, of the named entities in the
list for each passage.
2:Clearly resolve pronouns to their specific
names to maintain clarity.
Case Learning:
Inputs:
["Radio City","India","3 July 2001","Hindi","
English","May 2008", "PlanetRadiocity.com"]
Radio City
Radio City is India's first private FM radio
station and was started on 3 July 2001.
It plays Hindi, English and regional songs.
Radio City recently forayed into New Media in
May 2008 with the launch of a music portal -
PlanetRadiocity.com that offers music related
news, videos, songs, and other music-related
features.
Outputs:
{"triples": [
            ["Radio City", "located in", "India
"],
            ["Radio City", "is", "private FM
radio station"],
            ["Radio City", "started on", "3 July
 2001"],
            ["Radio City", "plays songs in", "
Hindi"],
            ["Radio City", "plays songs in", "
English"],
            ["Radio City", "forayed into", "New
Media"],
            ["Radio City", "launched", "
PlanetRadiocity.com"],
            ["PlanetRadiocity.com", "launched in
", "May 2008"],
            ["PlanetRadiocity.com", "is", "music
 portal"],
            ["PlanetRadiocity.com", "offers", "
news"],
            ["PlanetRadiocity.com", "offers", "
videos"],
            ["PlanetRadiocity.com", "offers", "
songs"]
    ]
}
```

Listing 3: Prompt for Questions Generating

```
Your task is to generate atomic questions
from a given triple (head, relation, tail) based
 on the provided Background Text.
Rules:
1.  Anchor to local context:
   First, locate the sentence(s) in the
Background Text that most directly express the
triple.
   Use details from the SAME local context (
nearby sentences), not random facts.
2. Richness requirement (MANDATORY):
   Each question MUST include at least TWO
distinct contextual anchors chosen from:
```

```
     (a) time/date/event
     (b) location
     (c) role/purpose
     (d) reception/outcome
   - Do NOT invent facts not in the Background
Text.
4. Coreference & canonical entity resolution (
MANDATORY, NO PRONOUNS AS ENTITIES):
   If head or tail is a pronoun or a generic/
underspecified mention,
     you MUST resolve it to the most specific
explicit entity name or explicit noun phrase
     stated in the Background Text near the
triple.
5. Atomic Questions: Generate EXACTLY 2
questions:
   Question 1: The question must contain <tail>.
 The answer must be the <head>.
   Question 2: The question must contain <head>.
 The answer must be the <tail>.
6. Define Edges:
   "answer_edge": The entity that is the correct
 answer to the question.
   "anchor_edge": The other entity from the
triple that serves as the context/clue.
7. Output Format: Output ONLY a JSON object.
```

Listing 4: Prompt for BiqRAG

```
Solve complex questions step by step.
You have two actions:
(1) SubQuestion[query] — Ask about ONE missing
fact. Get evidence.
(2) Finish[answer] — Return the final answer.
=== ATOMIC RULE (never break this) ===
Each SubQuestion must ask about exactly ONE fact.

- BAD (two facts in one question): SubQuestion[
Who scored the goal and won the match?]
- GOOD: SubQuestion[Who scored the goal?]
- GOOD: SubQuestion[Who won the match?]
If you find yourself writing "and", "or", or a
question with multiple conditions — STOP. Split
it into separate steps.
=== Examples ===
Question: Which team did the author of "The Old
Man and the Sea" play for?
Thought1: I need to first find out who wrote the
 book. Then find that person's team.
State Memory:
- Target: team of the book's author
- Known facts: none
Action1: SubQuestion[Who wrote "The Old Man and
the Sea"?]
Observation1: "The Old Man and the Sea" is a
novel written by Ernest Hemingway, published in
1952.
Thought2: Now I know the author is Ernest
Hemingway. I need to find which team he played
for.
State Memory:
- Author -> Ernest Hemingway
- Remaining: team of Hemingway
Action2: SubQuestion[Which sports team did
Ernest Hemingway play for?]
Observation2: Ernest Hemingway was not a
professional athlete. He was a known fan of the
New York Yankees.
Thought3: I now know the answer.
State Memory:
- Author -> Ernest Hemingway
- Hemingway -> fan of New York Yankees
Action3: Finish[New York Yankees]
Question: Which city is the university located
in that is affiliated with the laboratory where
the researcher who created the "Morningstar
System" works?
Thought1: I need to first identify who created
the Morningstar System. Then find their lab.
Then the lab's university. Then the university's
 city.
State Memory:
- Target: city of the university affiliated with
 the lab of the creator of Morningstar
- Known facts: none
Action1: SubQuestion[Who created the Morningstar
 System?]
Observation1: The Morningstar System was
developed by researcher Zhao Lan.
Thought2: Now I know Zhao Lan created it. I need
 her laboratory.
State Memory:
- Morningstar System -> Zhao Lan
- Remaining: Zhao Lan's laboratory, affiliated
university, city
Action2: SubQuestion[Which laboratory does Zhao
Lan work in?]
Observation2: Zhao Lan is a senior researcher at
 the Intelligent Computing Laboratory.
Thought3: Now I know the lab. I need the
affiliated university.
State Memory:
- Morningstar System -> Zhao Lan
- Zhao Lan -> Intelligent Computing Laboratory
- Remaining: affiliated university, city
Action3: SubQuestion[Which university is the
Intelligent Computing Laboratory affiliated with
?]
Observation3: The Intelligent Computing
Laboratory is affiliated with East China
University of Engineering.
Thought4: Now I know the university. I need the
city.
State Memory:
- Morningstar System -> Zhao Lan
- Zhao Lan -> Intelligent Computing Laboratory
- Intelligent Computing Laboratory -> East China
 University of Engineering
- Remaining: city of East China University of
Engineering
Action4: SubQuestion[In which city is East China
 University of Engineering located?]
Observation4: East China University of
Engineering is located in Hangzhou.
Thought5: I have all the information.
State Memory:
- Morningstar System -> Zhao Lan
- Zhao Lan -> Intelligent Computing Laboratory
- Intelligent Computing Laboratory -> East China
 University of Engineering
- East China University of Engineering ->
Hangzhou
Action5: Finish[Hangzhou]
```