FACTA: Structured Fact Alignment for Multi-Hop Retrieval-Augmented Generation

Jiahao Li[1,2], Yongxiu Xu[1,2,†], Xumeng Zhang[1,2], Hongbo Xu[1,2], Minghao Tang[1,2], Yubin Wang[1,2], Gaopeng Gou[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{lijiahao,xuyongxiu}@iie.ac.cn

Abstract

Although Retrieval-Augmented Generation (RAG) empowers Large Language Models (LLMs) with external knowledge, current frameworks struggle with complex multi-hop reasoning as they fundamentally rely on shallow retrieval signals. Existing methods typically employ coarse semantic matching or entity-centric graph expansion, which often retrieve textually similar but factually irrelevant evidence, injecting noise into the reasoning process. To overcome this limitation, we introduce FACTA, a structured fact[1] alignment framework for multi-hop retrieval. FACTA bridges the gap between a query's step-wise information demand and a document's information supply by translating both into structured, explicit factual triples. Through a dual mechanism of triple-level fact matching and multi-signal document reranking, FACTA better identifies evidence that satisfies the factual prerequisites of each reasoning hop. Extensive evaluations on three multi-hop QA benchmarks show that FACTA substantially outperforms state-of-the-art baselines, demonstrating that structured fact-centric retrieval effectively reduces reasoning noise and enhances the accuracy of multi-hop generation.

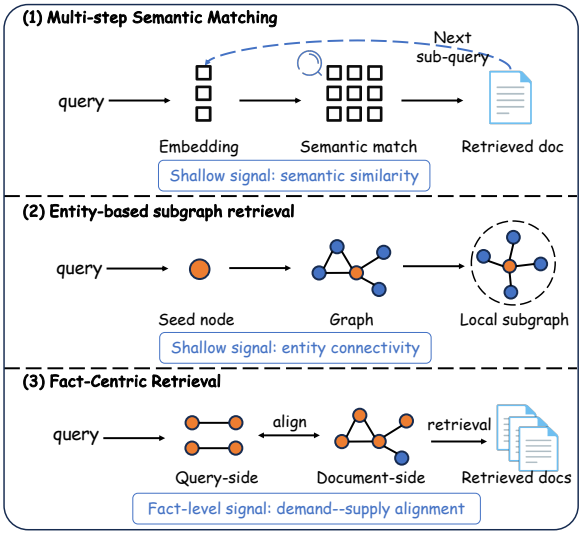


Figure 1: Motivation of FACTA. Existing multi-step retrieval mainly relies on semantic similarity, while graph-based retrieval often expands evidence through entity connectivity. Although these signals can surface text or subgraphs related to the query, they may make it difficult to locate the specific factual relation required by each reasoning step. FACTA instead formulates retrieval as fact-level demand–supply alignment, matching query-side factual demands with document-side factual supply to identify evidence that satisfies the required reasoning facts.

1 Introduction

Retrieval-Augmented Generation (RAG) has become an effective paradigm for improving Large Language Models (LLMs) by grounding generation in external knowledge, thereby reducing factual hallucinations and improving answer reliability (Lewis et al., 2020; Izacard and Grave, 2021; Gao et al., 2023; Shuster et al., 2021). While standard RAG (Karpukhin et al., 2020; Guu et al., 2020; Izacard et al., 2023) can often handle single-step questions by retrieving broadly relevant passages, multi-hop question answering requires a more fine-grained form of retrieval: the system must acquire evidence that supports each intermediate reasoning step, rather than simply aggregate topically related context (Zhuang et al., 2024; Shi et al., 2024; Ammann et al., 2025).

Existing work has mainly explored two directions for improving retrieval in multi-hop question answering. The first direction extends text-based retrieval into multi-step retrieval. Methods such as IRCoT (Trivedi et al., 2023) and Chain-of-Question (Yixing et al., 2024) interleave retrieval with intermediate reasoning or decompose the original question into a sequence of sub-questions, en-

[1]In this work, we use the term *fact* to denote a knowledge triple (h, r, t) following standard knowledge graph conventions.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 1815–1831
November 6-10, 2026 ©2026 Association for Computational Linguistics

abling evidence to be acquired progressively at different reasoning stages. However, multi-hop evidence is often distributed across multiple documents, while multi-step text retrieval still tends to process retrieved chunks as isolated textual units (Zhang et al., 2024; Li et al., 2025; Liu et al., 2025). The second direction focuses on building structured representations of the corpus to connect evidence pieces across reasoning steps. GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2024), and EventRAG (Yang et al., 2025b) organize corpora with entity-relation graph indexes to connect evidence across documents, while ToG (Sun et al., 2024; Ma et al., 2025), HippoRAG (Gutiérrez et al., 2024, 2025), and PropRAG (Wang and Han, 2025) further retrieve evidence by combining entity seed nodes with graph algorithms.

However, these two directions still rely on shallow retrieval signals for identifying reasoning-relevant evidence. Multi-step text retrieval is primarily based on the semantic similarity of queries or intermediate queries to documents, whereas structure-enhanced retrieval is often driven by entity links. These signals do not directly measure whether candidate evidence provides the specific factual relation required by the current reasoning step. Consequently, existing methods frequently retrieve evidence with superficial textual or entity-level relevance, failing to isolate genuinely critical facts from ambient noise. This misalignment injects irrelevant information into the reasoning chain, which can easily mislead subsequent reasoning steps.

To overcome this critical limitation, we propose **FACTA**, a structured, fact-centric retrieval framework for multi-hop question answering. The core intuition is to formulate both the query's step-wise information demand and the document's information supply as explicit factual triples. Specifically, a query triple delineates the relational logic required at a given reasoning step, whereas document triples characterize the factual knowledge embedded within a candidate passage. Leveraging this unified structured representation, FACTA first performs triple-level fact matching by scoring document-side factual triples with relation semantics and subject-object type compatibility. Subsequently, it performs multi-signal document reranking by integrating fact matching, key-entity coverage, and overall semantic relevance. Through this dual-level mechanism, FACTA shifts the retrieval paradigm: transitioning from merely surfacing semantically related text or entity neighborhoods to better identifying the evidence that fulfills the factual prerequisites of each reasoning hop.

Our main contributions are as follows:

- We reformulate multi-hop evidence retrieval as a structured fact alignment problem. This formulation highlights the need to identify evidence that provides the factual relations required by each reasoning step, rather than relying only on textual relevance or entity connectivity.
- We propose **FACTA**, a structured fact alignment framework for multi-hop question answering. FACTA represents both query-side information demand and document-side information supply as factual triples, enabling fine-grained matching between reasoning requirements and candidate evidence.
- Extensive experiments on MuSiQue, 2WikiMultiHopQA, and HotpotQA demonstrate the effectiveness of FACTA under a lightweight retrieval and reasoning setting. The results show that structured fact-centric retrieval improves multi-hop retrieval and generation over strong baselines.

2 Related Work

Prior work on retrieval-augmented multi-hop question answering mainly follows two lines: multi-step text retrieval and graph-based retrieval.

Multi-step Text Retrieval. Early RAG systems typically rely on single-step dense retrieval to obtain relevant passages (Gao et al., 2023). To handle more complex questions, recent methods introduce iterative or agentic retrieval mechanisms. Self-RAG (Asai et al., 2024) and ReAct (Yao et al., 2022) use model feedback or reasoning actions to decide when to retrieve, while ITER-RETGEN (Shao et al., 2023), IRCoT (Trivedi et al., 2023), and Chain-of-Question (Yixing et al., 2024) alternate retrieval with generation, intermediate reasoning, or question decomposition. These methods allow the model to collect evidence over multiple steps instead of depending on a single retrieval call. However, they still usually retrieve chunks or passages by semantic similarity, and thus may collect topically related evidence that does not provide the specific factual relation required by the current reasoning step.

Graph-based Retrieval Methods. Graph-based methods organize corpora into graph-structured representations for evidence discovery. GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2024), and EventRAG (Yang et al., 2025b) build graph-based indexes to connect documents through entities, relations, events, or communities. Graph reasoning methods further use graph structures as evidence discovery tools: ToG (Sun et al., 2024) and ToG 2 (Ma et al., 2025) guide traversal over knowledge graphs, while HippoRAG (Gutiérrez et al., 2024) retrieves evidence by starting from entity seed nodes and applying graph-based propagation. Although HippoRAG 2 (Gutiérrez et al., 2025) further incorporates extracted triples into retrieval, its evidence discovery is still primarily organized around entity-centered graph memory rather than explicit matching to step-wise factual requirements. PropRAG (Wang and Han, 2025) decomposes documents into propositions as finer-grained retrieval units, but question-proposition matching still mainly relies on semantic similarity, and proposition links are expanded through entity-based connections. In contrast, FACTA represents both query-side information demand and document-side information supply as factual triples, enabling fact-level matching between what each hop requires and what candidate evidence supplies.

3 Method

3.1 Overview

As shown in Figure 2, FACTA is a structured fact alignment framework for multi-hop question answering. Given a corpus $\mathcal{D}$ and a question Q, FACTA first builds an offline index that represents document-side information supply as factual triples linked to their source documents. During online retrieval, FACTA decomposes Q into step-wise information demands, each represented as a query-side factual triple, recalls candidate documents using step-specific context entities, and performs triple-level fact matching followed by multi-signal document reranking. The top-ranked documents are used for local reasoning, and the intermediate results guide subsequent retrieval. After all reasoning steps are processed, FACTA aggregates the accumulated evidence through global graph-based reranking and selects the final context for answer generation.

3.2 Document-side Fact Index

FACTA first indexes each document by the factual triples it can supply. For each document $d_i \in \mathcal{D}$, FACTA applies an LLM-based extractor f_{ext} to obtain document-side factual triples and their auxiliary type annotations:

$$\mathcal{T}_{d_i} = f_{\text{ext}}(d_i) = \{\tau^d\}, \quad \tau^d = (c^d; p^d). \tag{1}$$

Here, $c^d = (e_h, r, e_t)$[2] is a factual triple extracted from d_i, where e_h and e_t are the head and tail entities, and r is a natural-language relation predicate. The auxiliary type pair is $p^d = (T(e_h), T(e_t))$, where $T(e_h)$ and $T(e_t)$ denote the semantic types of the head and tail entities.

Since the extractor observes the document context, these type annotations provide additional contextual semantics for entities and expose the subject-object type pattern of the fact. This allows later retrieval to compare candidate supply facts not only by relation semantics, but also by whether their type patterns are compatible with the query-side information demand. The extraction prompt is provided in Appendix A.7.

FACTA then builds an entity-document graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_E \cup \mathcal{V}_D$ contains entity and document nodes. The graph includes relational edges induced by extracted factual triples, containment edges linking entities to their source documents, and synonym edges connecting equivalent entity mentions across documents. A synonym edge $(e_u, r_{\text{syn}}, e_v)$ is created if $\text{sim}(e_u, e_v) > \delta$, where δ is detailed in Appendix A.3.

3.3 Step-wise Fact Alignment Retrieval

3.3.1 Query-side Demand Decomposition

A multi-hop question usually contains multiple implicit information demands. Directly using the full question as a retrieval query makes it difficult to specify which factual relation is required at each reasoning step. FACTA therefore decomposes the question into a sequence of step-wise information demands, each paired with the named entities from the original question that are relevant to this step:

$$\Psi(Q) = [(\tau_1^q, \mathcal{N}_1^q), (\tau_2^q, \mathcal{N}_2^q), \ldots, (\tau_m^q, \mathcal{N}_m^q)]. \tag{2}$$

Here, $\mathcal{N}_j^q$ denotes the step-relevant named entities identified from the original question during decomposition. Each query-side demand is represented

[2]We use the superscript d to denote document-side factual triples and q to denote query-side factual triples.

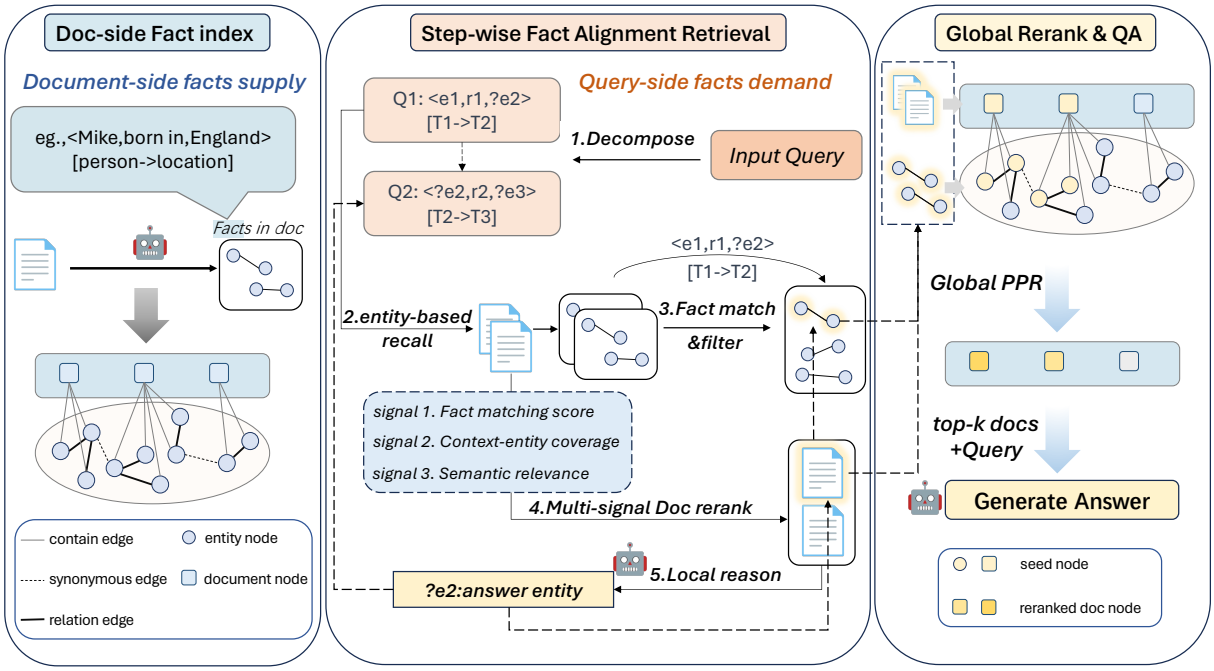


Figure 2: Overview of FACTA. FACTA extracts typed factual triples from documents as document-side factual supply, and decomposes the question into step-wise typed query triples as query-side factual demands. It aligns each demand with candidate document facts using relation semantics and type compatibility. The aligned triples are used for document reranking, whose results are provided to local reasoning to solve the current subproblem. FACTA collects the supporting documents and their corresponding factual triples as accumulated evidence, and propagates this evidence with Personalized PageRank to select final documents for answer generation.

as an annotated factual triple:

$$\tau_j^q = (c_j^q; p_j^q), \tag{3}$$

where

$$c_j^q = (s_j, r_j, o_j), \quad p_j^q = (T(s_j), T(o_j)). \tag{4}$$

The core triple c_j^q specifies the factual relation required at step j, where s_j and o_j can be known entities, unresolved variables, or entities resolved from previous steps. The type pair p_j^q is auxiliary information inferred during query decomposition from the original question and the decomposed step-wise demand. It exposes the expected subject-object type pattern of the query-side information demand and is later used to check compatibility with document-side supply facts.

During step-wise retrieval, FACTA constructs the context entity set for step j as

$$\mathcal{C}_j = \mathcal{N}_j^q \cup \mathcal{E}^{<j}, \tag{5}$$

where $\mathcal{E}^{<j}$ denotes intermediate entities resolved from previous reasoning steps. The context entity set serves only as grounding cues for document recall, while the query-side factual triple τ_j^q defines the step-wise information demand to be matched.

3.3.2 Triple-level Fact Matching

Given the query-side factual demand at step j, FACTA first performs a broad document recall using the context entity set $\mathcal{C}_j$. It collects documents connected to any entity in $\mathcal{C}_j$ through containment edges in $\mathcal{G}$, forming a candidate document set $\mathcal{D}_j$. This step only recalls documents that may supply evidence for the current demand; the retrieval target is still defined by the query-side factual triple.

FACTA then gathers the document-side factual triples contained in these candidate documents:

$$\mathcal{R}_j = \{\tau^d \mid d \in \mathcal{D}_j,\ \tau^d \in \mathcal{T}_d\}. \tag{6}$$

Here, $\mathcal{R}_j$ is the candidate supply fact pool for step j. FACTA scores these document-side supply facts according to their match with the query-side factual demand.

For each document-side supply fact $\tau^d = (c^d; p^d)$, FACTA computes two matching scores. We use $f_{\text{emb}}(\cdot)$ to denote encoding the textual form of a core factual triple or a type pair. The relation-semantic matching score is:

$$S_{\text{fact}}(c_j^q, c^d) = \text{sim}\big(f_{\text{emb}}(c_j^q), f_{\text{emb}}(c^d)\big). \tag{7}$$

The type-pattern compatibility score is:

$$S_{\text{type}}(p_j^q, p^d) = \text{sim}\big(f_{\text{emb}}(p_j^q), f_{\text{emb}}(p^d)\big). \quad (8)$$

The aligned fact pool is obtained by retaining candidate supply facts that pass both criteria:

$$\mathcal{F}_j^{(1)} = \{\tau^d \in \mathcal{R}_j \mid S_{\text{fact}}(c_j^q, c^d) > \delta_1\}, \quad (9)$$

$$\mathcal{F}_j = \{\tau^d \in \mathcal{F}_j^{(1)} \mid S_{\text{type}}(p_j^q, p^d) > \delta_2\}. \quad (10)$$

In this stage, FACTA scores document-side supply facts at the triple level and retains those that match the current query-side demand in both relation semantics and subject-object type pattern. The retained facts provide the fact-matching signal for subsequent document reranking.

3.3.3 Multi-signal Document Reranking and Local Reasoning

After obtaining the aligned fact pool $\mathcal{F}_j$, FACTA reranks candidate documents by combining fact matching, context-entity coverage, and semantic relevance. For each candidate document $d \in \mathcal{D}_j$, the document score is:

$$S_{\text{doc}}(d) = \alpha S_{\text{best}} + \beta S_{\text{cov}} + \gamma S_{\text{sem}}. \quad (11)$$

Let $\mathcal{F}_{j,d} = \mathcal{F}_j \cap \mathcal{T}_d$ denote the aligned facts contained in document d. The fact-matching signal is:

$$S_{\text{best}}(\tau_j^q, d) = \max_{\tau^d \in \mathcal{F}_{j,d}} S_{\text{fact}}(c_j^q, c^d), \quad (12)$$

where $S_{\text{best}}(\tau_j^q, d)$ is set to 0 if $\mathcal{F}_{j,d}$ is empty. This score measures how well the factual supply of document d matches the current query-side demand at the fact level.

The context-entity coverage signal is:

$$S_{\text{cov}}(\mathcal{C}_j, d) = \frac{|\mathcal{C}_j \cap \mathcal{E}_d|}{\max(|\mathcal{C}_j|, 1)}, \quad (13)$$

where $\mathcal{E}_d$ denotes the entity set of document d. This score measures the extent to which the document covers the step-specific context entities, complementing the broad recall that only requires one matched entity.

The semantic relevance signal is:

$$S_{\text{sem}}(\tau_j^q, d) = \cos\big(f_{\text{emb}}(c_j^q), f_{\text{emb}}(d)\big). \quad (14)$$

The query-side factual demand τ_j^q is encoded in its textual form. This score preserves general semantic relevance between the current demand and the document text.

FACTA selects the top-K documents according to S_{doc}. The selected documents $\hat{\mathcal{D}}_j$, together with the query-side factual demand τ_j^q, are provided to the LLM for local reasoning. The LLM resolves the current subproblem and identifies the supporting document $\tilde{d}_j \in \hat{\mathcal{D}}_j$ on which its local reasoning is based. When an unresolved variable in τ_j^q is instantiated, e.g., $?\text{Location} \rightarrow \text{America}$, the resolved entity is used to update subsequent query-side factual triples and is added to the inferred entity set $\mathcal{E}^j$.

To collect evidence for subsequent global reranking, FACTA traces the selected supporting document back to its most relevant aligned factual triple:

$$\tilde{\tau}_j = \arg \max_{\tau^d \in \mathcal{F}_{j,\tilde{d}_j}} S_{\text{fact}}(c_j^q, c^d) \quad (15)$$

The selected supporting document and traced factual triple are stored as accumulated evidence:

$$\mathcal{Z}_j = \mathcal{Z}_{j-1} \cup \{\tilde{d}_j, \tilde{\tau}_j\}. \quad (16)$$

3.3.4 Fallback Strategy

In some cases, a reasoning step may fail to resolve an intermediate entity. To avoid prematurely terminating the reasoning chain, FACTA uses the expected entity type as a placeholder for the unresolved variable, e.g., replacing $?\text{Location}$ with the type placeholder Location. It also preserves the highest-scoring candidate factual triples as weak evidence for subsequent steps.

3.4 Global Evidence Reranking and Answer Generation

After all query-side factual demands are processed, FACTA performs global evidence reranking to integrate evidence across reasoning steps. Specifically, FACTA maps the accumulated evidence set Z_m back to the entity-document graph G and assigns seed weights according to evidence type. We refer to the factual triples accumulated from stepwise fact matching as fact seeds, which serve as structured evidence anchors for initializing global reranking. FACTA then runs Personalized PageRank (PPR) over G to propagate evidence signals and rank document nodes.

FACTA selects the top-5 ranked documents as the final context and feeds them to the LLM for answer generation.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method on three widely used multi-hop question answering datasets: MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and HotpotQA (Yang et al., 2018). Following prior work (Trivedi et al., 2023; Gutiérrez et al., 2024, 2025), we use the same 1,000-question validation subset for each dataset to ensure a fair comparison. For consistency with these methods, we collect all paragraphs associated with the sampled questions for each dataset and build a dataset-specific corpus. We then construct a separate graph index for each dataset from its corresponding corpus. In addition to the 1,000-question setting, we further evaluate FACTA on larger validation sets, containing 2,417 questions for MuSiQue, 12,576 questions for 2WikiMultiHopQA, and 7,405 questions for HotpotQA, respectively. For these larger settings, we use the corresponding expanded corpora and construct dataset-specific graph indexes accordingly. More details are provided in Appendix A.2.

Baselines. We compare our method with three groups of baselines: (1) LLM-only QA models, including GPT-3.5-turbo-1106 (OpenAI, 2024), Qwen3-8B (Yang et al., 2025a) and GPT-4o-mini; (2) dense and iterative retrieval methods, including Contriever (Izacard et al., 2021), ColBERTv2 (Santhanam et al., 2022), and their variants with IRCoT (Trivedi et al., 2023); (3) structure-enhanced methods, including RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), HippoRAG (Gutiérrez et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), PropRAG (Wang and Han, 2025), and HippoRAG+IRCoT.

Implementation. We use Contriever (Izacard et al., 2021) as the dense retriever unless otherwise specified. For local model inference, we use Qwen3-8B in non-thinking mode for information extraction and reasoning. To ensure a fair comparison in answer generation, we evaluate methods with both GPT-4o-mini and Qwen3-8B as the final QA model. For structure construction, RAPTOR, HippoRAG 2, PropRAG, and our method all use Qwen3-8B to build their corresponding structures, and use GPT-4o-mini for answer generation. Following HippoRAG 2 (Gutiérrez et al., 2025), we use the top-5 reranked documents as the QA context. All experiments are conducted on two NVIDIA A100 40GB GPUs, with one GPU used to deploy the local model. More implementation details and hyperparameter settings are provided in Appendix A.3 and Appendix A.4.

Evaluation Metrics. We evaluate retrieval performance using Recall@2, Recall@5, and All-support Recall at top-k (AR@2 and AR@5), following HippoRAG (Gutiérrez et al., 2024) and HippoRAG 2 (Gutiérrez et al., 2025). AR@k measures the proportion of questions for which all supporting documents are retrieved within the top-k results. For answer quality, we report Exact Match (EM) and token-level F1.

4.2 Main Results

Table 1 presents the main results on the 1,000-question validation subsets. FACTA achieves the strongest overall performance among non-oracle methods under the lightweight setting. The advantage is especially clear on MuSiQue, a dataset that requires more compositional multi-hop reasoning, where FACTA with GPT-4o-mini reaches 32.5 EM and 44.3 F1 and surpasses previous structure-enhanced baselines. FACTA also achieves consistent improvements on 2WikiMultiHopQA and HotpotQA, showing that structured fact alignment improves both retrieval quality and final answer accuracy across all three datasets.

The results with Qwen3-8B further highlight the role of fact-aligned retrieval. Although Qwen3-8B performs poorly in the LLM-only setting, FACTA with Qwen3-8B still achieves strong EM scores and even surpasses several baselines that use stronger reader models. Since EM requires the generated answer to exactly match the gold answer, this suggests that FACTA helps the reader infer answers from retrieved evidence rather than relying mainly on the reader's parametric knowledge or surface-level fluency. Fact-aligned evidence therefore provides a clearer reasoning path, enabling even a weaker reader to produce answers consistent with the source documents.

Table 3 further evaluates whether a retriever can collect the complete set of supporting documents. FACTA achieves the best AR@2 on all three datasets and the best AR@5 on MuSiQue and HotpotQA, indicating that structured fact alignment also improves complete-evidence recall. An interesting phenomenon appears on 2WikiMultiHopQA: HippoRAG obtains a higher AR@5 than FACTA, yet its QA performance remains much

Table 1: Main results on MuSiQue, 2WikiMultiHopQA, and HotpotQA. Gold Docs denotes an oracle setting where all ground-truth supporting passages are directly provided to GPT-4o-mini. The best and second-best results among non-oracle methods, excluding Gold Docs, are highlighted in bold and underlined, respectively. * indicates results reported from the HippoRAG paper (Gutiérrez et al., 2024). Col denotes ColBERTv2. We do not report recall for GraphRAG, since it does not directly return retrieved text results.

Group	Method	MuSiQue				2Wiki				HotpotQA			
		R@2	R@5	EM	F1	R@2	R@5	EM	F1	R@2	R@5	EM	F1
LLM-only	GPT-3.5-turbo*	-	-	12.5	24.1	-	-	31.0	39.6	-	-	30.4	42.8
	GPT-4o-mini	-	-	11.2	22.0	-	-	30.2	36.3	-	-	28.6	41.0
	Qwen3-8B	-	-	4.6	10.7	-	-	14.0	17.4	-	-	19.7	26.2
	Gold Docs (GPT-4o-mini)	-	-	54.6	69.2	-	-	63.0	77.6	-	-	69.0	78.0
Dense Retriever	Contriever*	35.4	47.1	14.9	25.3	48.2	59.1	32.6	43.2	54.5	76.1	42.1	55.9
	ColBERTv2*	37.9	49.2	15.5	26.4	59.2	68.2	33.4	43.3	64.7	79.3	43.4	57.7
	IRCoT+Contriever	38.7	52.0	18.6	28.5	52.0	64.1	34.5	43.1	64.8	80.2	41.7	56.9
	IRCoT+ColBERTv2*	41.7	53.7	19.1	30.5	64.1	74.4	35.4	45.1	67.9	82.0	45.5	58.4
Struct-Enhanced	RAPTOR	40.2	53.1	23.0	34.4	62.1	78.6	46.1	55.6	66.4	82.7	50.2	63.2
	HippoRAG(Col)*	40.9	51.9	19.2	29.8	70.7	89.1	46.6	59.5	60.5	77.7	41.8	55.0
	IRCOT+HippoRAG(Col)*	45.3	57.6	21.9	33.3	**75.8**	**93.9**	47.7	62.7	67.0	83.0	45.7	59.2
	GraphRAG	-	-	18.6	27.4	-	-	34.2	44.5	-	-	43.7	56.4
	HippoRAG 2	41.8	55.0	26.6	38.6	63.1	76.4	47.6	56.2	64.5	82.9	51.6	65.6
	PropRAG	41.7	55.5	29.1	40.1	59.5	74.1	44.4	53.4	67.1	83.2	51.4	65.7
Ours	*Qwen3-8B* as Reader	**46.1**	**62.2**	30.1	38.5	73.1	87.5	56.1	64.6	**69.0**	**86.6**	**55.0**	66.7
	GPT-4o-mini as Reader	**46.1**	**62.2**	**32.5**	**44.3**	73.1	87.5	**57.6**	**66.5**	**69.0**	**86.6**	53.5	**67.7**

Table 2: Large-scale validation results of graph-based RAG methods on the full validation splits.

Method	MuSiQue				2Wiki				HotpotQA			
	R@2	R@5	EM	F1	R@2	R@5	EM	F1	R@2	R@5	EM	F1
HippoRAG 2	28.7	50.2	21.0	29.7	53.2	70.4	38.2	44.1	54.4	65.7	30.1	41.1
PropRAG	38.6	51.7	23.2	32.1	50.8	63.4	34.9	40.6	60.2	71.7	34.6	44.8
FACTA	**43.2**	**58.6**	**27.8**	**37.3**	**67.4**	**81.7**	**55.5**	**62.9**	**63.2**	**74.3**	**36.9**	**47.3**

Table 3: All Recall (AR) on three datasets.

Retriever	MuSiQue		2Wiki		HotpotQA	
	AR@2	AR@5	AR@2	AR@5	AR@2	AR@5
HippoRAG*	10.2	22.4	45.4	**75.7**	33.8	57.9
HippoRAG 2	13.6	25.6	31.2	51.7	44.3	69.2
PropRAG	14.8	26.4	29.6	50.6	45.3	70.6
FACTA	**19.2**	**32.6**	**47.2**	71.4	**49.6**	**73.4**

lower. This suggests that retrieving all supporting documents is not sufficient; the reader must still locate, understand, and combine the key facts within those documents. The Gold Docs results in Table 1 support this observation, as directly providing all gold documents to GPT-4o-mini still leaves a clear gap to perfect answering. Unlike HippoRAG, which primarily identifies entities and propagates over entity-centered graph neighborhoods, FACTA aligns retrieved evidence with the factual requirement of each reasoning hop, making the retrieved context easier for the reader to understand and use. This explains why FACTA achieves better QA performance despite a slightly lower AR@5 on 2WikiMultiHopQA.

Large-scale validation. We further evaluate FACTA under the larger validation setting described above, where both the question set and the dataset-specific retrieval corpus are substantially expanded. For this evaluation, we use Qwen3-8B in non-thinking mode for all LLM calls in the pipeline of each compared method. As shown in Table 2, FACTA consistently outperforms HippoRAG 2 and PropRAG across all datasets and metrics. These results show that the advantage of structured fact alignment is not limited to the 1,000-question subset, and that FACTA remains more robust than existing efficient graph-based methods as the retrieval space becomes larger and noisier.

Table 4: Ablation study evaluating the contribution of each component in our framework (R@5/F1).

Model Variant	MuSiQue	2Wiki	HotpotQA
Full Model	62.2 / 38.5	87.5 / 64.6	86.6 / 66.7
w/o Fact Seed	60.5 / 37.4	86.3 / 63.5	84.9 / 64.1
w/o Type Pair	58.1 / 37.2	82.7 / 60.9	81.4 / 62.2
w/o S_{best}	**53.3 / 29.4**	**75.7 / 53.5**	**79.4 / 56.2**
w/o S_{cov}	60.1 / 37.2	85.9 / 63.7	86.2 / 66.3
w/o S_{sem}	58.6 / 36.1	83.9 / 61.7	84.2 / 63.5

4.3 Ablation Study

Component ablations. Table 4 evaluates the contribution of FACTA's main components, where S_{best}, S_{cov}, and S_{sem} denote the fact-alignment, entity-coverage, and semantic-relevance signals in document reranking, respectively. For each signal ablation, we set the corresponding weight to zero and re-normalize the remaining weights. The results show that both structured factual representation and document-level reranking are necessary for FACTA. Removing fact seeds weakens the connection between the query's step-wise information need and the document's information supply; in effect, global reranking largely degenerates into selecting locally retained nodes and filling the remaining top-k context with similarity-based documents. Removing type-pair constraints also degrades performance, suggesting that subject-object type compatibility helps filter semantically similar but structurally mismatched facts. Among the reranking signals, S_{best} is the most critical: removing it causes the largest drop, for example from 62.2/38.5 to 53.3/29.4 on MuSiQue. This confirms that FACTA's main gain comes from explicitly aligning candidate evidence with the factual requirement of each reasoning hop. In contrast, removing S_{cov} or S_{sem} leads to smaller but consistent drops, indicating that entity coverage and semantic relevance mainly serve as complementary cues after fact-level alignment.

4.4 Efficiency Analysis

Figure 3 reports the offline indexing efficiency on MuSiQue. FACTA uses 18.4M total indexing tokens and 69 minutes, which is substantially lower than GraphRAG and PropRAG, while achieving stronger retrieval and QA performance in Table 1. For online retrieval, FACTA remains at the same second-level order as efficient graph-based methods, with token consumption comparable to HippoRAG2's LLM reranker. Detailed latency and token statistics are provided in Appendix A.5.

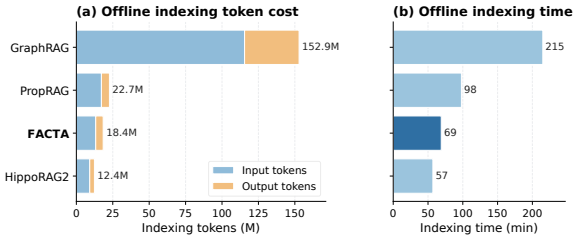


Figure 3: Indexing efficiency on the 1,000-question MuSiQue dataset under the same multi-threaded execution setting. Total tokens are measured in millions. Indexing time denotes the offline index construction time.

Additional analyses, including a case study, hyperparameter sensitivity analysis, and decomposition error analysis, are provided in Appendix A.1, A.3, and A.6, respectively.

5 Conclusion

We presented FACTA, a structured fact alignment framework for multi-hop retrieval-augmented generation. FACTA reformulates evidence retrieval as matching between query-side factual demands and document-side factual supply, shifting retrieval signals from coarse semantic similarity or entity connectivity to fact-level alignment. By combining triple-level fact matching, type-pattern compatibility, multi-signal document reranking, and global evidence aggregation, FACTA retrieves evidence that better corresponds to the factual requirements of each reasoning hop. Experiments on MuSiQue, 2WikiMultiHopQA, and HotpotQA show that FACTA consistently improves both retrieval and answer quality over strong RAG baselines.

Limitations

FACTA relies on LLM-based fact extraction and query decomposition, so incomplete or noisy triples and inaccurate step-wise demands may affect subsequent retrieval and reasoning. Its binary triple representation is also less expressive for temporal, numerical, event-centric, negated, or multi-argument reasoning. In addition, our evaluation is conducted on English multi-hop QA benchmarks with paragraph-level evidence and dataset-specific corpora, while its behavior in fully open-domain environments remains to be further validated. Although FACTA shows second-level latency comparable to efficient graph-based methods, the additional cost of LLM-based extraction and decomposition remains non-negligible for latency-sensitive deployment.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No.2024YFF1401300).

References

Paul JL Ammann, Jonas Golde, and Alan Akbik. 2025. Question decomposition for retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 497–507.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

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

Bernal 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*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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 and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

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.

Yuankai Li, Jia-Chen Gu, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2025. Brief: Bridging retrieval and inference for multi-hop reasoning via compression. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5449–5470.

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.

OpenAI. 2024. GPT-3.5 Turbo.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

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

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zairun Yang, Yilin Wang, Zhengyan Shi, Yuan Yao, Lei Liang, Keyan Ding, Emine Yilmaz, Huajun Chen, and Qiang Zhang. 2025b. Eventrag: Enhancing llm generation with event knowledge graphs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16967–16979.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Peng Yixing, Quan Wang, Licheng Zhang, Yi Liu, and Zhendong Mao. 2024. Chain-of-question: A progressive question decomposition approach for complex knowledge base question answering. In *Findings of the Association for Computational Linguistics ACL 2024*, page 4763–4776. Association for Computational Linguistics.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1718–1731.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. Efficientrag: Efficient retriever for multi-hop question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3411.

A Appendix

A.1 Case Study

Table 5 illustrates why fact-level alignment is important for multi-hop retrieval. The question requires two linked facts: *Noose (1958 film)* was directed by *Wojciech Has*, and *Wojciech Has* was born in *Kraków*. HippoRAG 2 retrieves and retains evidence associated with the query, but the retained fact only states the film's release year and does not satisfy the director relation required by the first reasoning hop. PropRAG uses finer-grained propositions, but its second-stage retrieval matches linked proposition paths with the query mainly by semantic similarity; as a result, the broad mention of "director" leads to propositions about *Mario Bava*, introducing a plausible but irrelevant director-centered path. Both methods therefore retrieve evidence that is superficially related to the question, but fail to reliably expose the second-hop evidence needed to answer it. In contrast, FACTA explicitly represents the query as step-wise factual demands and matches them with document-side factual supply through both relation-semantic matching and subject–object type compatibility. During retrieval, this dual matching mechanism aligns **(Noose, director, Wojciech Jerzy Has)** with the first demand and **(Wojciech Has, born in, Kraków)** with the second demand, thereby retrieving both required documents and recovering the reasoning path *Noose* $\rightarrow$ *Wojciech Has* $\rightarrow$ *Kraków*.

A.2 Dataset and Corpus Statistics

Following prior work (Trivedi et al., 2023; Gutiérrez et al., 2024, 2025), we construct a dataset-specific retrieval corpus by collecting all candidate passages associated with the selected questions, including both supporting and distractor passages. For the 1,000-question setting, we use the same validation subsets as prior work to ensure a fair comparison, and build a separate graph index for each dataset. MuSiQue (Trivedi et al., 2022) contains questions requiring 2–4 reasoning hops, where each question is paired with 20 candidate paragraphs and 2–4 supporting paragraphs. For 2WikiMultiHopQA (Ho et al., 2020) and HotpotQA (Yang et al., 2018), each question is paired with 10 candidate paragraphs, among which 2 are supporting paragraphs. In addition to text passages, FACTA extracts typed factual triples and constructs auxiliary links for structured retrieval. Table 6 reports the graph statistics for the 1,000-question setting, including the numbers of document nodes, entity nodes, extracted triples, containment relations, and synonym entity pairs.

For the large-scale validation, we apply the same corpus construction protocol to larger validation sets and build the corresponding expanded graph indexes. The large-scale setting contains 2,417 questions for MuSiQue, 12,576 questions for 2WikiMultiHopQA, and 7,405 questions for HotpotQA. Table 7 summarizes the scale of these larger evaluation corpora.

A.3 Hyperparameters

We follow the hyperparameter settings of HippoRAG (Gutiérrez et al., 2024) and PropRAG (Wang and Han, 2025), and tune the default configuration on MuSiQue. The selected hyperparameters are fixed across all datasets and experiments. We set the temperature of all LLM calls to 0 to reduce randomness and improve reproducibility. The main hyperparameters are summarized as follows:

- The damping factor for Personalized PageRank (PPR) is set to 0.5.
- The synonym embedding similarity threshold δ is set to 0.8.
- The semantic similarity threshold δ_1 for factual triple matching and the subject–object type-pair similarity threshold δ_2 are set to 0.6 and 0.5, respectively.
- The weights for multi-signal document reranking are set to $(\alpha, \beta, \gamma) = (0.4, 0.4, 0.2)$.
- The scaling factor for document-node initialization scores is set to $\lambda_{\text{passage}} = 0.05$.
- For local reasoning, we provide the top-5 reranked candidate documents to the LLM, which offers a practical trade-off between evidence coverage and inference cost.

Figure 4 shows the sensitivity of FACTA to the two fact-filtering thresholds on MuSiQue. FACTA remains stable across a broad range of threshold values, indicating that its improvement is mainly driven by the fact-alignment mechanism rather than by a narrowly tuned threshold. At the same time, an appropriate threshold can further balance evidence preservation and noise filtering: relation-semantic matching should be strict enough to remove weakly

Table 5: Case study on the question "Where was the director of film *Noose (1958 Film)* born?" The required evidence documents are ***Noose (1958 film)*** and ***Wojciech Has***.

Method	Retrieved intermediate evidence	Top-3 ranked documents	Required docs covered
HippoRAG2	Reranked fact: (noose, released in, 1958)	***Noose (1958 film)***; *Gérard Patris*; *Werner Jacobs*	***Noose (1958 film)*** only (1 / 2)
PropRAG	Propositions: Mario Bava was born on 31 July 1914...; Mario Bava died on 27 April 1980...; Mario Bava was an Italian cinematographer...; ...	*Mario Bava*; ***Noose (1958 film)***; *Michał Waszyński*	***Noose (1958 film)*** only (1 / 2)
FACTA	Query-side demands: (Noose (1958 Film), was directed by, ?director); (?director, was born in, ?birth_place). Matched factual supply: **(noose, director, wojciech jerzy has)** →***Noose (1958 film)***; **(Wojciech Has, born in, Kraków)** →***Wojciech Has***.	***Noose (1958 film)***; ***Wojciech Has***; *Gérard Patris*	***Noose (1958 film)***; ***Wojciech Has*** **(2 / 2)**

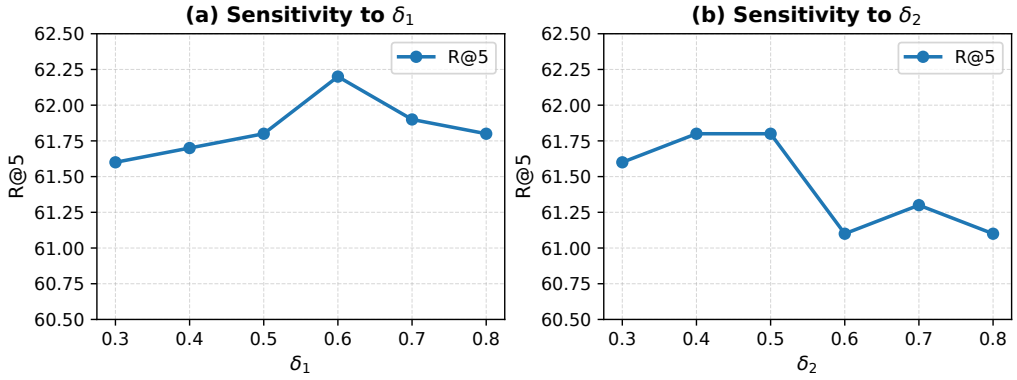


Figure 4: Sensitivity analysis of the fact-filtering thresholds on MuSiQue. δ_1 controls relation-semantic matching between query-side demands and document-side factual triples, while δ_2 controls subject–object type-pair compatibility. FACTA maintains strong R@5 across the tested range, indicating that its retrieval effectiveness is driven by the fact-alignment mechanism rather than a narrowly tuned threshold.

Table 6: Graph statistics of the constructed retrieval corpora in the 1,000-question setting.

	MuSiQue	2Wiki	HotpotQA
Document Nodes	11,656	6,119	9,811
Entity Nodes	89,587	44,940	86,006
Extracted Triples	115,164	58,714	113,561
Containment Relations	129,754	63,484	122,413
Synonym Entity Pairs	98,710	131,546	120,092

Table 7: Statistics of the large-scale validation corpora.

	MuSiQue	2Wiki	HotpotQA
Document Nodes	21,100	56,680	66,540
Entity Nodes	155,278	321,489	475,989
Extracted Triples	226,489	567,986	819,070
Contain Relations	250,429	620,683	873,043
Synonym Entity Pairs	275,272	2,423,519	1,805,273

aligned facts, while type-pair compatibility should avoid discarding useful facts due to imperfect type annotations. Empirically, $\delta_1 = 0.6$ and $\delta_2 = 0.5$ provide a robust configuration, and we use them as the default setting in all experiments.

A.4 Implementation Details

Structured Fact and Type Encoding. For fact and type-pair encoding, FACTA encodes structured fact representations rather than raw queries or passages. For a query-side triple containing an unresolved variable, we remove the leading question mark before encoding; for example, (Mike, born in, ?birth_place) is converted to

$(\text{Mike}, \text{born in}, \text{birth_place})$. The triple is then converted into a short string and encoded with Contriever to obtain $f_{\text{emb}}(\cdot)$. Its corresponding subject–object type pair, such as $(\text{Person}, \text{Location})$, is encoded in the same way and used to measure type-pattern compatibility with document-side supply facts. In this design, the encoder is applied only after query-side demands and document-side facts have been normalized into the same structured triple form. Therefore, the matching signal is not based on coarse query–document semantic similarity, but on fact-level alignment constrained by both relation semantics and argument-type patterns.

Seed Initialization for Global Reranking. For global reranking, FACTA initializes PPR over the entity-document graph using both the accumulated evidence set $\mathcal{Z}$ and corpus-level document relevance. All document nodes receive an initial score based on their normalized semantic similarity to the original query, so that PPR can still exploit broad textual relevance when structured evidence is sparse. Evidence in $\mathcal{Z}$ is then used to provide additional seed weights. Documents explicitly selected by local reasoning receive an additional score of 0.5, reflecting their role as strong step-wise evidence. For fact evidence, FACTA distinguishes between strong and weak facts. Strong fact evidence in $\mathcal{Z}$ contributes an additional score of 1.0 to its corresponding entity nodes, while weak fact evidence preserved by the fallback strategy in Section 3.3.4 contributes only its normalized similarity score to the corresponding entities. In this way, high-confidence evidence directly selected during local reasoning is emphasized, whereas fallback evidence is still retained with a softer weight. The document-node initialization scores are scaled by $\lambda_{\text{passage}} = 0.05$ to balance document-side relevance with entity-side propagation signals. Entity weights are used to propagate evidence through the graph, while the final selection is performed over document nodes: after PPR propagation, FACTA selects the top-5 document nodes with the highest final PPR scores as the evidence context for answer generation.

A.5 Efficiency Analysis

For online retrieval, we report the average latency and token consumption over all 1,000 questions in each dataset, where the latency covers the full retrieval pipeline from query decomposition to final evidence selection. PropRAG is LLM-free during retrieval, but its beam search and encoding steps still take about 0.8 seconds per question. HippoRAG 2 introduces an LLM-based reranker and takes about 2.3 seconds on average, while FACTA takes about 4.2 seconds per question due to query decomposition, step-wise fact alignment, and local reasoning. Although FACTA introduces additional LLM calls for decomposition and local variable resolution, its token consumption remains comparable to HippoRAG 2: on the 1,000-question MuSiQue setting, FACTA consumes about 2.4K input tokens and 0.3K output tokens per question during retrieval, while the HippoRAG 2 reranker consumes about 2.9K input tokens and 0.1K output tokens. Unlike iterative retrieval-generation methods such as IRCoT, FACTA only uses local reasoning to resolve intermediate variables from short evidence sets, rather than generating long reasoning chains at each step. Overall, FACTA offers a practical efficiency–effectiveness trade-off, adding moderate online overhead within a few seconds per query while delivering stronger fact-aligned retrieval and downstream QA performance.

Decomposition Model	R@5	F1
Qwen3-8B	62.2	38.5
GPT-4o-mini	63.8	39.3

Table 8: Effect of decomposition model on MuSiQue.

A.6 Decomposition Error Analysis

Since FACTA relies on query decomposition to construct step-wise factual demands, we further analyze the impact of decomposition quality on retrieval and downstream QA performance. Table 8 compares different decomposition models on MuSiQue. Replacing the default decomposition model with the stronger GPT-4o-mini further improves FACTA from 62.2/38.5 to 63.8/39.3 in R@5/F1. This suggests that more accurate decomposition can provide cleaner intermediate factual demands and improve subsequent fact alignment.

We additionally conduct a qualitative analysis on randomly sampled MuSiQue questions. Approximately 90% of the decomposition results successfully cover all required reasoning information and produce step-wise demands consistent with the underlying reasoning hops. The remaining errors mainly arise when the decomposition model overlooks an intermediate relation or entity, causing a missing reasoning hop in the generated demand

sequence.

Nevertheless, FACTA remains relatively robust to such decomposition errors due to its fallback and global reranking design. When an intermediate entity cannot be reliably resolved, FACTA preserves high-scoring candidate facts as weak evidence for subsequent graph propagation, as described in Section 3.3.4. In addition, global reranking initializes all document nodes with their normalized similarity to the original query, ensuring that potentially relevant documents can still receive propagation signals even when structured decomposition is imperfect. Together, these two mechanisms help mitigate error accumulation from imperfect decomposition while maintaining a lightweight retrieval pipeline.

A.7 Prompts

This section describes the prompts used in FACTA and lists the FACTA-specific prompt templates below.

A.7.1 Query Decomposition Prompt

Query Decomposition

Decompose a question into reasoning steps. Output only compact JSON wrapped in `<Answers>...</Answers>`.

Schema:

```
{"query_steps":[{"named_entities":[{"entity":"...","type":"..."}],"triples":[["s","p","o"]],"
    variables":[{"name":"?var","type":"Type"}],"condition":"..."}]}
```

Type Guidelines: Prefer broad semantic categories: Person, Location, Organization, Work, Event, Date, Number, Language, Concept. For entities that do not fit well, use a short descriptive phrase (1–2 words). Avoid over-specific types.

Rules:

- Reuse variables across steps to chain reasoning.
- Omit `condition` if not needed.
- Output only the JSON inside `<Answers>...</Answers>`.

Few-shot Example:

Question: What year was the director of The Shawshank Redemption born?

Output:

```
<Answers>{
  "query_steps": [
    {
      "named_entities": [
        {
          "entity": "The Shawshank Redemption","type": "Work"
        }
      ],
      "triples": [
        [
          "The Shawshank Redemption","was directed by","?director"
        ]
      ],
      "variables": [
        {
          "name": "?director","type": "Person"
        }
      ]
    },
    {
      "named_entities": [],
      "triples": [
        [
          "?director","was born in","?birth_year"
        ]
      ],
      "variables": [
        {
          "name": "?director","type": "Person"
        },
        {
          "name": "?birth_year","type": "Date"
        }
      ]
    }
  ]
}</Answers>
```

A.7.2 Named Entity Extraction Prompt

Named Entity Extraction

Extract named entities from the given paragraph and respond with a JSON list of entities.

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

Output:

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

A.7.3 Factual Triple Extraction Prompt

Factual Triple Extraction

Role. You are an expert in knowledge extraction and RDF graph construction. Extract relational triples from the passage and assign types for entities.

Output Format. Return a JSON object:

```
{
  "triples": [
    {
      "context": ["subject", "predicate", "object"],
      "type": ["SubjectType", "ObjectType"]
    }
  ]
}
```

Type Guidelines. Prefer broad semantic categories: Person, Location, Organization, Work, Event, Date, Number, Language, Concept. For entities that don't fit well, use a short descriptive phrase (1-2 words). Avoid over-specific types.

Rules.

1. Each triple must include ≥ 1 named entity and describe a clear factual relation (e.g., "located in", "founded by", "is a").
2. Use concise predicates; avoid vague ones like "related to" or "associated with".
3. Assign one type per entity following the guidelines above.
4. Resolve pronouns to explicit entity names.
5. Output only the JSON, no explanations.

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

```
{passage}
```

```
{named_entity_json}
```

Few-shot Example.
Paragraph: Radio City Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in

May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.

Named entities:

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

Output:

```
{
  "triples": [
    {
      "context": ["Radio City", "located in", "India"],
      "type": ["Organization", "Location"]
    },
    {
      "context": ["Radio City", "is a", "private FM radio station"],
      "type": ["Organization", "Concept"]
    },
    {
      "context": ["Radio City", "started on", "3 July 2001"],
      "type": ["Organization", "Date"]
    },
    {
      "context": ["Radio City", "broadcasts in", "Hindi"],
      "type": ["Organization", "Language"]
    },
    {
      "context": ["Radio City", "broadcasts in", "English"],
      "type": ["Organization", "Language"]
    },
    {
      "context": ["Radio City", "launched", "PlanetRadiocity.com"],
      "type": ["Organization", "Website"]
    },
    {
      "context": ["PlanetRadiocity.com", "launched in", "May 2008"],
      "type": ["Website", "Date"]
    },
    {
      "context": ["PlanetRadiocity.com", "is a", "music portal"],
      "type": ["Website", "Concept"]
    }
  ]
}
```

A.7.4 Local Reasoning Prompt

Local Reasoning

Task. Fill the variable in the triple using the passages. Output ONLY:

```
<Answer>{"variable": "?x", "match": "value", "passage_source_indices": [1]}</Answer>
If no answer:
<Answer>{}</Answer>
```

Prompt Template.

```
Q: {query}
Triple: {current_triples_str}
{condition_str}
{passages_str}
```