# CONRAG: LIGHTWEIGHT INFERENCE OF MULTI-HOP RELATIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Understanding how two entities are connected often requires tracing multi-hop relations across documents to identify intermediate entities and supporting evidence that explain a connection. This is a task that appears frequently in scientific research and other knowledge-intensive analyses. We formalise this setting as multi-hop relation inference: given two known endpoint entities, we aim to recover the bridge entities and evidence-grounded reasoning chains that connect them across a document corpus, and to generate an explanation grounded in the retrieved evidence. Existing multi-hop RAG systems typically seek an unknown answer entity rather than explicitly recovering the connection between two known endpoints and graph-based approaches often rely on costly LLM-extracted knowledge graphs that limit scalability to large document collections. We introduce CONRAG, which builds a lightweight entity–document graph from entity co-occurrence and LLM-based entity filtering. Its connective retrieval infers and semantically ranks paths between two endpoints. On MuSiQue and 2WikiMultiHopQA, CONRAG consistently improves bridge entity and reasoning chain recovery over strong RAG baselines, while reducing graph-indexing token cost by up to roughly 1.5 orders of magnitude. Our results show that endpoint-constrained path retrieval provides an effective and index-efficient approach to evidence-grounded relation discovery.

## 1 INTRODUCTION

Retrieval Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2023) enables language models to access and reason over large document collections without requiring the entire corpus to fit within the input context. Rather than relying solely on knowledge encoded in model parameters, RAG systems retrieve a subset of relevant external information and provide it to a Large Language Model (LLM) as additional context for generation. By incorporating information not contained in the training corpus, RAG mitigates hallucinations by grounding generation in external knowledge (Shi et al., 2023; Liu et al., 2024). This improves answer quality and performance across knowledge-intensive natural language tasks (Gao et al., 2023; Fan et al., 2024).

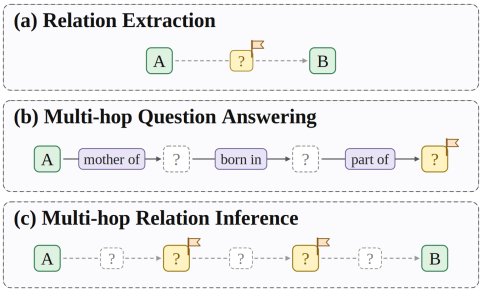


Figure 1: Relation extraction, MHQA and multi-hop relation inference.

**From simple relations to chain retrieval.** Extracting entities and their relations from documents is important for real-world applications such as knowledge discovery (Peng et al., 2017), contextualisation (Edge et al., 2024; Guo et al., 2025), scientific research and question answering (Fader et al., 2014). These relations have been studied in different forms (Niklaus et al., 2018; Han et al., 2020). Some are stated explicitly, such as "Caesar was born in Rome" (Zhang et al., 2017). Others are inferred by combining stated relations, for example when two parent links imply a grandparent relation (Yao et al., 2019; 2021; Cabot & Navigli, 2021). Prior work targets this challenge, including multi-hop relation extraction across documents (Wang et al., 2022; Lu et al., 2023; Son et al., 2023). These fall under the task of relation extraction as shown in Figure 1 *(a)*. *(b)* and *(c)* involve more complex connections that may only be expressed through a chain of intermediate entities. A researcher may be linked to a company through a co-author who



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

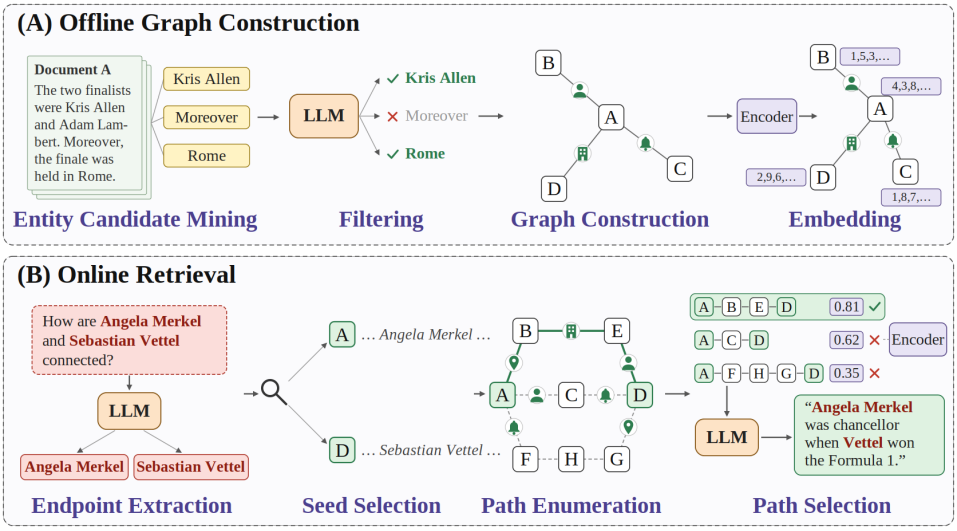


Figure 2: **Overview of ConRAG** (outlined in Section 3). **(A) Offline graph construction:** An LLM filters entity candidates mined from the documents, the remaining entities link documents in a graph and a dense encoder embeds each document. **(B) Online retrieval:** Anchor and target are extracted from the question. Seed documents are retrieved for both. Paths are enumerated and scored by question-document similarity. The top path documents form the context to explain the relation.

founded a startup later acquired by that company. Multi-hop question answering (MHQA) (Mavi et al., 2024) (Figure 1*(b)*) involves answering questions like *"Where was the author of Marcinkus born?"* where these relations and an anchor entity *(Marcinkus)* are given in the question (Trivedi et al., 2022; Ho et al., 2020; Yang et al., 2018). We instead treat the chain itself and not a single relation label as the target: Given two known entities, we retrieve the bridge entities and links connecting them (Figure 1 *(c)*). We call this *multi-hop relation inference*.

**Why existing multi-hop RAG does not solve it.** As outlined, MHQA requires reasoning across multiple steps (Mavi et al., 2024). State-of-the-art RAG methods for MHQA either interleave retrieval with stepwise LLM reasoning (Trivedi et al., 2023; Sun et al., 2024) or retrieve over an LLM-extracted knowledge graph (KG) (Gutierrez et al., 2024; Gutiérrez et al., 2025). Despite this, they are not explicitly designed to reconstruct the chains connecting two given entities. Their retrieval objectives are typically relevance-based or answer-oriented rather than endpoint-constrained: Documents may be scored individually or graph neighborhoods expanded toward an unknown answer, without requiring the evidence to form a connected path between two known endpoints. Consequently, essential bridge documents with weak query similarity may be missed.

**Can multi-hop relation inference be natively supported in RAG?** To close the gap and enable RAG based multi-hop relation inference, we introduce Connective Retrieval Augmented Generation (ConRAG)[1], a graph-based RAG framework designed to recover multi-hop connections between two known entities. ConRAG constructs an entity-document graph from a corpus of documents and retrieves paths between endpoint-associated documents, surfacing the intermediate entities and evidence needed to explain their connection. To evaluate this setting, we give the standard MHQA benchmarks MuSiQue (Trivedi et al., 2022) and 2WikiMultiHopQA (Ho et al., 2020) a new interpretation. Question and answer entities become the known endpoint pairs, while annotated bridge entities and decomposition steps become the target chains. ConRAG outperforms RAG baselines in bridge entity recovery and improves reasoning chain recall and precision, while reducing indexing token cost by up to 1.5 orders of magnitude. More precisely, our contributions are the following:

- We formalise multi-hop relation inference and introduce new benchmarks derived from MuSiQue and 2WikiMultiHopQA, with known endpoint pairs, bridge entities, and reasoning chains as targets. This way, we provide a testbed and recipe for future research on relation discovery.
- We introduce ConRAG, the first graph-based RAG framework designed to recover connections between known entities, with a token efficient indexing solution via LLM entity filtering.
- We show that ConRAG outperforms all evaluated RAG baselines on the studied benchmarks.

[1]Code available at: `https://anonymous.4open.science/r/ConRAG_iclr/README.md`

## 2 Related Work

**Explaining relations.** Explaining how two entities are related, rather than merely detecting that they are, has a long history across data management, the Semantic Web and NLP (Anyanwu & Sheth, 2003; Faloutsos et al., 2004; Swanson, 1986), motivated by exploratory search and by users' needs for supporting evidence in high-stakes domains (Bhatia et al., 2018; Pirrò, 2015). Works like REX (Fang et al., 2011) infer and rank explanatory paths for relevance over pre-existing KGs, while Voskarides et al. (2015) combine KG information with sentences retrieved from a document corpus to explain a relation. In contrast, our method operates on raw text, induces its own entity-document graph, and uses an LLM to explain the recovered relation chains in prose.

**Relation extraction.** A broader line of work infers direct relations between entities within a sentence or document (Zhao et al., 2024). DocRED (Yao et al., 2019) extends this to document-level extraction, and models such as REBEL (Cabot & Navigli, 2021) and UIE (Lu et al., 2022) generate relation triplets end-to-end. Moreover, CodRED (Yao et al., 2021) moves to the cross-document setting, where the relation between an entity pair must be inferred from evidence spread over several documents. Graph-based methods such as MR.COD (Lu et al., 2023), a multi-hop evidence retriever for cross-document relation extraction, and PILOT (Son et al., 2023), construct entity-document paths and train dense retrievers to select evidence for a downstream classifier that predicts a predefined relation label between endpoints. Path inference over KGs follows a similar formulation. From path-ranking random walks (Lao et al., 2011) to reinforcement-learning walkers (Xiong et al., 2017), multi-hop paths over curated triples serve as evidence for predicting a direct relation between two given entities. Importantly, these methods are not framed as RAG systems: Retrieved paths serve as evidence for a relation extractor or classifier, rather than as context for an LLM to generate an evidence-grounded explanation. Therefore, in these settings the output is a direct relation label, and intermediate entities and documents are auxiliary. Our setting inverts this: The label is dropped, and the chain of bridge entities and evidence becomes the target. ConRAG introduces the relation inference setting within RAG and builds on the evidence-path retrieval formulation of Lu et al. (2023), while removing its reliance on gold mention annotations and a task-specific trained retriever in favour of an efficient, LLM-filtered entity vocabulary.

**Retrieval Augmented Generation.** RAG (Lewis et al., 2020) conditions generation on documents retrieved from an external corpus. It was proposed for knowledge-intensive tasks and later adopted to reduce unsupported claims by LLMs and to inject knowledge absent from pretraining (Gao et al., 2023; Fan et al., 2024). Standard pipelines retrieve documents by semantic similarity. Graph-based RAG addresses this by organising the corpus as a graph. GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2025) extract entities and relations with an LLM and retrieve over the resulting KG. PathRAG (Chen et al., 2026) alters this by pruning paths for retrieval. MiniRAG (Fan et al., 2025) builds a lightweight graph of chunks and extracted entities. LinearRAG (Zhuang et al., 2026) bypasses relation extraction altogether, connecting SpaCy-extracted entities through shared sentences and documents and retrieving by propagating activation from the entities mentioned in the query. More cost-efficient RAGs are outlined in Appendix A. ConRAG is closest to MiniRAG and LinearRAG in using relation-free entity-document graph and to PathRAGs retrieval. It differs in the following respects. First, the entity vocabulary is inferred from corpus co-occurrence statistics and LLM filtering. Second, compared to PathRAG, retrieval is not pruning based and traverses a graph without relations. We search for paths between two known endpoints in a construction-efficient graph and rank path candidates on average document similarity with an encoder.

**Multi-hop reasoning.** MHQA benchmarks such as HotpotQA, 2WikiMultiHopQA, and MuSiQue (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022) require composing facts across documents. Some MHQA questions contain an anchor entity and a sequence of relations leading to an answer through unknown bridge entities. Existing methods that address this challenge fall into several families. Iterative retrieval interleaves retrieval with LLM reasoning as in IRCoT (Trivedi et al., 2023), or lets an LLM traverse a KG (Sun et al., 2024). The resulting chain from the question's anchor to its answer is a byproduct. Chain-aware retrievers such as MDR (Xiong et al., 2021) and CogQA (Ding et al., 2019) learn to retrieve sequences of documents or to expand an entity graph from the question, to reach an unknown answer. Finally, graph-based methods like HippoRAG and HippoRAG 2 (Gutierrez et al., 2024; Gutiérrez et al., 2025) leverage graph algorithms to surface important relational information. None of these families explicitly targets the setting of two known entities with an unlabeled, evidence-grounded chain between them. ConRAG makes the chain of bridge entities and evidence the output itself, and assumes no predefined relation vocabulary.

# 3 CONRAG: CONNECTIVE RETRIEVAL AUGMENTED GENERATION

We propose CONRAG, a RAG framework for multi-hop relation inference that improves inference and understanding of indirect relations between two entities.

## 3.1 PROBLEM FORMULATION

Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ be a corpus of $N$ documents. A question $q$ names an *anchor* entity $a$ and a *target* entity $b$ like *Rome* and *Pythagoras*. The system must produce output $\mathcal{O}(q)$

$$\mathcal{O}(q) = (\mathcal{C}, y), \qquad \mathcal{C} = \big(u_1 \to u_2,\ \ldots,\ u_{k-1} \to u_k\big), \tag{1}$$

where $\mathcal{C}$ is an ordered sequence of $k$ entities with $u_1 = a$ and $u_k = b$, written as directed links. The links outline the indirect relation between $a$ and $b$. $y$ is a prose explanation of the indirect relation. The interior entities are the *bridges*. Both $\mathcal{C}$ and $y$ must be entailed by a retrieved document set $\mathcal{R}(q) \subseteq \mathcal{D}$. Emitting $\mathcal{C}$ separately from $y$ makes the claimed reasoning path directly confirmable rather than inferable from free text.

## 3.2 ENTITY QUALIFICATION

To construct the bipartite entity-document graph for retrieval, we first derive an entity vocabulary from document texts which then serves as a connector between documents. CONRAG proceeds in the two steps below.

**Entity candidate mining.** Two families of candidates are extracted from each document. The first is the set of lowercased and possessive stripped unigrams. The second is the set of phrases of at most $n_{\max}$ tokens drawn from maximal sequences of capitalised tokens. For each candidate $e$ we record the document set $\mathcal{D}_e$ containing it and its document frequency $\mathrm{df}(e) = |\mathcal{D}_e|$ over the whole corpus. Candidates with $\mathrm{df}(e) < 2$ are discarded since a mention confined to one document cannot connect two documents.

**LLM-based entity filtering.** Mined candidates are filtered by an LLM with instructions given in Appendix E.2 containing the capitalisation rate of the candidates across the corpus. It labels genuine named entities and discards other candidates in containment-aware batches. Candidates linked by contiguous token containment (*united* $\subset$ *united states* $\subset$ *united states of america*) are placed in the same batch of size $B$. Competing granularities of one name are thus adjudicated in a single context rather than in independent calls.

## 3.3 OFFLINE INDEXING

Building upon the entity vocabulary we construct an index used for the retrieval of relevant information during inference as shown in Figure 2 *(A)*. The index is a triple $\mathcal{I} = (\mathcal{D}, G, \mathbf{V})$: The corpus $\mathcal{D}$, a bipartite entity-document graph $G$, and a document embedding matrix $\mathbf{V} \in \mathbb{R}^{N \times r}$, where $r$ is the embedding dimension and whose row $\mathbf{v}_d$ holds the vector of document $d$. Before indexing, $\mathcal{D}$ is normalised by deduplicating pairs of titles and texts and collapsing whitespaces from documents in the datasets.

This relation-free construction avoids committing to a predefined relation schema and eliminates the need to extract relation triples from every document. Entities act only as connectors between documents, while the original documents remain the source of evidence presented to the generator.

**Graph construction.** To construct the graph, entities are first assigned to documents from the incidence recorded during mining (Sec. 3.2). Moreover, if one entity is a contiguous token subsequence of another within a document, only the longer is kept. In this way a document is linked by the most specific name available to it. Based on this, we construct an undirected bipartite graph $G = (\mathcal{V}_\mathcal{D} \cup \mathcal{V}_\mathcal{E}, E)$ where $\mathcal{V}_\mathcal{D}$ is the set of document nodes and $\mathcal{V}_\mathcal{E}$ is the set of entity nodes. An undirected edge exists if and only if an entity is assigned to a document. Together, all edges form the set of edges $E$. There are no document-document or entity-entity edges.

**Document embedding.** Each document is embedded from its title and text with a dense encoder to give $\mathbf{V}$. These vectors serve two roles downstream, dense seeding and query-document support.

**Node canonicalisation.** Entity nodes are canonicalised across the corpus. For this purpose entity strings from the vocabulary are embedded as well. Candidate pairs for canonicalisation are restricted to entities that share a word or whose initials are acronyms of the other. Nodes among these pairs whose cosine similarity exceeds a threshold $\theta$ are merged by the union-find algorithm (Galler & Fisher, 1964; Tarjan, 1975), and each group is represented by its member with the highest document frequency. Since entity embeddings are not used for retrieval they do not become part of $\mathbf{V}$.

### 3.4 ONLINE RETRIEVAL

During retrieval CONRAG aims to collect information that is meaningful to reconstruct indirect connections between entities.The retrieval procedure is summarised in Algorithm 1 and illustrated in Figure 2 *(B)*.

**Algorithm 1** CONRAG Online Retrieval

**Require:** question $q$, graph $G$, embeddings $\mathbf{V}$, corpus $\mathcal{D}$
**Ensure:** retrieved context $\mathcal{R}(q)$
1: Extract anchor $a$ and target $b$ from $q$
2: Retrieve dense seeds $\mathcal{R}_{\text{seed}}(q)$
3: Ensure coverage of both endpoints
4: Enumerate paths between endpoint-side seeds
5: Score paths by question-document similarity
6: **if** no valid path is found **then**
7: Apply shortest-path search
8: **end if**
9: Select top paths as $\mathcal{R}_{\text{path}}(q)$
10: **return** $\mathcal{R}_{\text{seed}}(q) \cup \mathcal{R}_{\text{path}}(q)$

**Endpoint extraction.** Entity mentions are extracted from question $q$ by an LLM. The first two mentions are taken as the anchor $a$ and the target $b$. If fewer than two mentions are found, $q$ itself is used as the seeding signal and the endpoint constraint is skipped.

**Seeding.** Seed documents are retrieved densely. Documents are ranked by $\cos(\mathbf{v}_q, \mathbf{v}_d)$, where $\mathbf{v}_q$ is the embedding of the whole question and $\mathbf{v}_d$ is a document embedding. The top $k_{\text{seed}}$ documents form $\mathcal{R}_{\text{seed}}(q)$. A path from $a$ to $b$ requires seed documents covering both endpoints. Top-$k$ dense retrieval can return one-sided seed sets, thus we restrict $\mathcal{R}_{\text{seed}}(q)$ to documents containing $a$ or $b$ and inject a matching document from the corpus when an endpoint is missing from the semantic search. If this fails, we fall back to a search with the entity embedding against the documents. Finally, if this does not yield documents containing an entity, we use the top-$k_{\text{fallback}}$ documents of the question embedding search for the missing endpoint.

**Path enumeration.** Following Lu et al. (2023), we enumerate paths by depth-first search over $d \to e \to d'$ transitions from the anchor-side seeds. A path of $L$ documents is

$$P = \big((d_1, \ldots, d_L), (e_1, \ldots, e_{L-1})\big), \qquad e_i \text{ the entity shared by } d_i \text{ and } d_{i+1}, \tag{2}$$

with all $d_i$ distinct and $L \le L_{\max}$, where $L_{\max}$ is the maximum path length in documents. Entities whose degree $\deg(e) = |\mathcal{D}_e|$ exceeds a $p_{\text{hub}}$ quantile of the degree distribution are removed from the traversal adjacency. Enumeration is constrained. A path is retained only if node $d_L$ lies in the target-side seed pool.

**Path selection.** We score paths by the mean query support of documents as in Eq. 3 (Lu et al., 2023). Scored paths are bucketed by length. The $k_{\text{path}}$ highest-scoring paths are retained for each length in a target set $\mathcal{L}$, and their documents form $\mathcal{R}_{\text{path}}(q)$. If no enumerated path has a length in $\mathcal{L}$, retrieval falls back to a single shortest path. The candidate with the highest score under Eq. 3 is emitted alone. The retrieval procedure is bounded by the maximum path length $\mathcal{L}_{\max}$, while hub suppression limits expansion through high-degree entities. Together, these constraints keep connective search tractable while the shortest path fallback preserves connectivity when the pruned graph is too sparse.

$$S(P) = \frac{1}{L} \sum_{i=1}^{L} \cos(\mathbf{v}_q, \mathbf{v}_{d_i}). \tag{3}$$

**Context assembly.** The generator LLM receives $\mathcal{R}(q) = \mathcal{R}_{\text{seed}}(q) \cup \mathcal{R}_{\text{path}}(q)$. Seeds supply endpoint evidence and remain available when the graph is locally disconnected. Paths supply the intermediate documents. Which documents carry the relations is left to the generator.

### 3.5 CONSTRUCTING MULTI-HOP RELATION INFERENCE BENCHMARKS

Multi-hop question answering datasets with question decompositions $S$ can be repurposed for multi-hop relation inference. Given an MHQA question-answer pair with a linear reasoning chain, we

identify the entity from which the decomposition starts as the *anchor* $a$ and use the original answer as the *target* $b$. If the anchor is not explicitly given in the decomposition, we extract from its first decomposition question with an LLM instructed by a prompt given in Appendix E.1.

For an $h$-hop decomposition where $h$ is the number of reasoning hops, the intermediate answers define the ordered bridge entities $Z = [z_1, \ldots, z_{h-1}]$. Together with the anchor and target, they form the gold relation chain consisting of intermediate links.

$$c^\star = (a \to z_1 \to \cdots \to z_{h-1} \to b).$$

We then replace the original question with a relation question $q$ that names only the two endpoints, e.g., "How are $a$ and $b$ related?". The resulting example consists of the endpoint pair $(a, b)$, the bridge entities $Z$, the gold chain $c^\star$, and the original supporting documents $\mathcal{D}_{\text{gold}}$. We restrict the construction to linear chains with a single anchor, since branching questions require recovering multiple interacting relation chains. Algorithm 2 summarises the general procedure.

**Algorithm 2** MHQA to Multi-hop Relation Inference

**Require:** MHQA decomposition $S$, answer $b$, supporting documents $\mathcal{D}_{\text{gold}}$
**Ensure:** relation-inference example $(q, a, b, Z, c^\star, \mathcal{D}_{\text{gold}})$
1: Identify anchor $a$ from the first step of $S$
2: Collect intermediate answers $Z = [z_1, \ldots, z_{h-1}]$
3: Set answer $b$ as the target entity
4: Build chain $c^\star = (a \to z_1 \to \cdots \to z_{h-1} \to b)$
5: Form relation question $q$ from endpoints $(a, b)$
6: **return** $(q, a, b, Z, c^\star, \mathcal{D}_{\text{gold}})$

## 4 Experimental Setup

**Datasets.** We instantiate the benchmark construction procedure from Section 3.5 on the validation sets of MuSiQue (Trivedi et al., 2022) and 2WikiMultiHopQA (Ho et al., 2020). For MuSiQue, we retain the linear *2hop*, *3hop1*, and *4hop1* compositions. This yields 2,066 examples which comprise 1,252 two-hop, 568 three-hop, and 246 four-hop questions. For 2WikiMultiHopQA, we retain the *compositional* and *inference* categories, yielding 6,785 two-hop examples. Retrieval operates over pooled, deduplicated validation corpora rather than per-question context bundles, comprising approximately 26.3k documents for MuSiQue and 56.7k for 2WikiMultiHopQA. Dataset-specific processing and anchor extraction details are provided in Appendix B.

**Evaluation.** We introduce three derived metrics for multi-hop relations benchmarking, *Bridge contains match accuracy (BCM)*, *chain recall (CR)* and *chain precision (CP)*. BCM measures the share of gold bridge entities included in a response. CR and CP measure the share of gold links recovered, as well as the share of generated links that are gold. We additionally evaluate response quality against the baselines. We follow LightRAG (Guo et al., 2025) and use their prompt to instruct an LLM judge comparing ConRAG's responses against baselines in terms of *comprehensiveness*, *diversity*, *empowerment* and *overall* response quality. Finally, we report ConRAG's average win rate against each baseline. For quantitative results, we evaluate three 1,000-question samples drawn with different seeds. While limiting the question corpus to reduce generation cost, we still use the full document corpus of the validation set for retrieval to show indexing efficiency.

**Baselines** We compare ConRAG against four RAG baselines. We include *Naïve RAG* (Lewis et al., 2020) as a semantic search baseline. Moreover, ConRAG is compared against *LightRAG* as being one of the most capable graph models regarding qualitative metrics (Guo et al., 2025). Third, we include *PathRAG* since it uses a path-based retrieval similar to ConRAG (Chen et al., 2026). Finally, *HippoRAG 2* is included as a strong multi-hop reasoning baseline (Gutiérrez et al., 2025). With this setup, we compare ConRAG against methods that leverage relation-rich KGs, which can provide fine-grained information about the underlying corpus for generation. Baseline implementation details are outlined in Appendix B.2.

**Implementation Details.** All methods use OpenAI's *text-embedding-3-small* model for semantic embedding. HippoRAG 2, LightRAG, and PathRAG use *GPT-4o-mini* for graph indexing. ConRAG uses *GPT-4o* for entity filtering and generation. The values of ConRAG's parameters are listed in Appendix B Table 5 and kept fixed across datasets. All methods use the same generation prompt (Appendix E.3), with method-specific retrieved context. As the number of retrieved documents from ConRAG averages to 10.8 we limit the retrieval of naïve RAG, HippoRAG and LightRAG to the top 10 documents as well. PathRAG retrieval varies and is not restrictable and therefore remains as is. We use *GPT-4o-mini* as LLM judge for qualitative criteria.

# 5 RESULTS

Our evaluation is organised around four questions: *(i)* Does CONRAG infer multi-hop relations more accurately than traditional and graph-based RAG baselines? *(ii)* Does this translate into higher-quality responses as judged by a qualitative evaluation? *(iii)* How much of the gain stems from retrieval alone, independent of LLM-based generation? *(iv)* How does CONRAG's efficiency compare in terms of indexing and inference cost to the baselines?

## 5.1 RELATION INFERENCE

Table 1: Bridge Contains Match Accuracy (BCM), Chain Recall (CR), and Chain Precision (CP) in % on MuSiQue and 2WikiMultiHopQA (mean ± std over 3 seeds). Best per column in **bold**, second-best underlined.

| | *MuSiQue* | | | *2WikiMultiHopQA* | | |
|---|---|---|---|---|---|---|
| Method | *BCM* | *CR* | *CP* | *BCM* | *CR* | *CP* |
| NaïveRAG | 59.2±0.6 | 19.6±0.4 | 27.4±1.1 | 61.6±1.5 | 33.4±2.4 | 40.1±2.3 |
| LightRAG | 58.0±1.7 | 20.2±0.3 | 28.3±0.1 | 72.1±0.2 | 40.2±1.0 | 35.7±1.1 |
| PathRAG | 58.3±0.8 | 17.2±0.3 | 33.6±1.3 | 64.9±0.2 | 30.4±1.5 | 35.8±1.8 |
| HippoRAG 2 | 59.6±2.2 | 23.3±0.6 | **36.4±1.7** | 67.0±1.0 | 40.2±0.9 | 34.9±2.1 |
| CONRAG | **71.2±0.7** | **29.5±0.5** | 34.3±1.1 | **81.0±1.4** | **49.4±1.3** | **48.1±0.9** |

In Table 1 we present results regarding performance on relation inference. It is worth to highlight that MuSiQue poses a more challenging task for relation inference since it contains relations with up to 3 intermediate steps, while 2WikiMultiHopQA only contains relations with one intermediate step for inference. The results allow to make the following observations:

**Superior performance on multi-hop relation inference.** Due to its mechanism which is designed to connect entities, CONRAG achieves results that underpin its ability to surface relations. Most prominently, this is visible through BCM which highlights a methods awareness regarding the existence of bridge entities. CONRAG outperforms baselines across both datasets. Moreover, CONRAG proves strong ability to infer reasoning chain links. Deltas over 6.2 points on MuSiQue and 9.2 points on 2WikiMultiHopQA in chain recall to HippoRAG 2 indicate that even against systems laid out for precise multi-hop reasoning, CONRAG is capable to infer and understand relations more effectively.

**Longer relation chains remain challenging.** As shown in Figure 3, we can derive that with rising hop count, inference becomes more difficult. Performance decreases substantially with hop count across all methods. Nevertheless, CONRAG retains the highest bridge-recovery at every hop count (Appendix C Table 7), outperforming the strongest baseline . Overall, this can be seen as a result of an increase in possible chain connections with a rise in hopcount due to more combinatorial options for path assembly. When controlling for hop count, CONRAG consistently performs better than baselines across both benchmarks, indicating performance robustness. While all models show decreasing performance over hop count, we can also observe that baseline performance varies strongly between the two-hop questions of the different datasets. As outlined in Appendix C Table 7, deviations for baseline models rise up to 10.6 points for Naïve RAG on BCM and up to 9.9 points for chain recall (LightRAG).

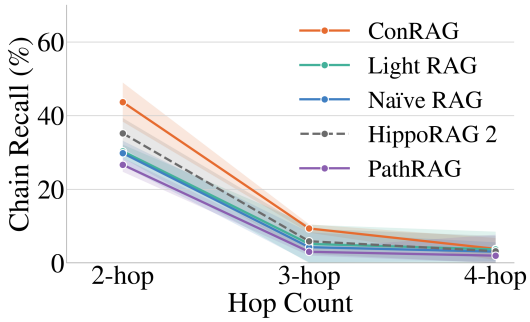


Figure 3: **Decreasing chain recall (%).** All models show lower chain recall as the number of reasoning hops increases. Bands indicate two-sided 99% t-confidence intervals.

**Ranking of baselines varies.** Notably, our experiments show that baseline models do not keep a consistent ranking across metrics and datasets among each other. Only HippoRAG 2 shows a

tendency in being the strongest baseline in chain recall, which indicates that its capabilities for multi-hop reasoning are partly transferable to multi-hop relation inference. For BCM and chain precision, there are no consistent top performers among the baseline models. In contrast, CONRAG consistently performs above baseline level.

### 5.2 QUALITATIVE PERFORMANCE

Table 2: **Pairwise qualitative win rate (%) of CONRAG against each baseline.** (reported as mean ± 95% Wilson half-width) For every question and criterion, the LLM judge selects either CONRAG or the baseline as the winner. Therefore, the two win rates sum to 100%, and only CONRAG's share is reported. Values above 50% indicate that CONRAG wins the majority of comparisons and are shown in **bold**. Compr. = comprehensiveness, Divers. = diversity, and Empow. = empowerment.

| CONRAG vs. Baseline | *MuSiQue* | | | | *2WikiMultiHopQA* | | | |
|---|---|---|---|---|---|---|---|---|
| | Compr. | Divers. | Empow. | Overall | Compr. | Divers. | Empow. | Overall |
| Naïve RAG | **54.3 ± 3.1** | **50.1 ± 3.1** | **53.5 ± 3.1** | **53.5 ± 3.1** | **57.7 ± 3.1** | 49.2 ± 3.1 | **57.2 ± 3.1** | **57.2 ± 3.1** |
| LightRAG | **69.6 ± 2.9** | **65.2 ± 3.0** | **69.6 ± 2.9** | **69.6 ± 2.9** | **60.1 ± 3.0** | **55.4 ± 3.1** | **59.8 ± 3.1** | **59.8 ± 3.1** |
| PathRAG | **63.1 ± 3.0** | **60.3 ± 3.0** | **62.0 ± 3.0** | **62.0 ± 3.0** | **62.1 ± 3.0** | **55.0 ± 3.1** | **61.7 ± 3.0** | **61.8 ± 3.0** |
| HippoRAG 2 | **65.8 ± 3.0** | **64.2 ± 3.0** | **65.5 ± 3.0** | **65.7 ± 3.0** | **61.7 ± 3.0** | **52.1 ± 3.1** | **61.0 ± 3.0** | **61.1 ± 3.0** |

We measure qualitative performance of CONRAG by reporting win rates on relation questions against baselines over one question sample for each dataset. In Table 2 we note win rates across both datasets and all four baselines. CONRAG attains the higher overall win rate. Naïve RAG generates the second most valuable answers in terms of our metrics and even outperforms CONRAG on diversity for 2WikiMultiHopQA. Counterintuitively, the win margin widens against graph-based RAGs. Most clearly, CONRAG wins more than twice as much against LightRAG over the MuSiQue dataset. This suggests that the improvement is not merely an effect of retrieving more but also less distracting context. Where LightRAG and PathRAG provide comprehensive generation prompts (Table 3) important details can get lost. CONRAG's retrieval works differently. Answers are drawn from a tighter, higher-precision evidence set, which trades breadth of stated content for relevance.

### 5.3 INDEXING AND INFERENCE EFFICIENCY

To design a RAG which is useful in a setting with large document corpora and regular requests, efficiency in terms of token cost for construction and generation has to be considered. CONRAG is designed to reduce token consumption during indexing to a minimum. Our results in Table 3 show that CONRAG reduces indexing cost in comparison to LightRAG and PathRAG on MuSiQue and 2WikiMultiHopQA by roughly 1.5 orders of magnitude. Furthermore, CONRAG also clearly undercuts the comparatively lightweight HippoRAG 2.

Table 3: Average prompt tokens (± std) & index construction tokens. **Bold** indicates lowest cost among graph-based RAGs.

| Model | *MuSiQue* | | *2WikiMultiHopQA* | |
|---|---|---|---|---|
| | Prompt | Construction | Prompt | Construction |
| PathRAG | 10.312±3.547 | 263.25 M | 11.546±3.494 | 564.37 M |
| LightRAG | 7.484±2.786 | 263.25 M | 8.427±3.031 | 564.37 M |
| Naïve RAG | **2.242±344** | – | **2.542±699** | – |
| HippoRAG 2 | 2.245±349 | 29.71 M | 2.854±833 | 65.27 M |
| CONRAG | 2.261±983 | **3.90 M** | 3.301±1.707 | **9.63 M** |

Moreover, CONRAG saves tokens during generation. Context heavy RAGs like LightRAG and PathRAG consume more tokens by design to provide valuable context for answers regarding qualitative metrics. CONRAG instead provides more compact context for the single purpose of surfacing and understanding multi-hop relations. It thereby attains a comparable generation token footprint as HippoRAG 2 which is designed for compact context retrieval to provide correct MHQA responses. Figure 5 in Appendix D illustrates the source of CONRAG's indexing efficiency. The document-frequency and LLM filtering stages remove most candidate entities while largely preserving the bridge entities required for path retrieval. This produces a substantially smaller entity vocabulary before graph construction.

### 5.4 RETRIEVAL

Finally, we isolate retrieval from generation to showcase improvement independent of the generator: Table 4 reports document recall, i.e. how much gold evidence reaches the generator before any answer is produced. Since CONRAG retrieves 10.8 documents on average, we report R@10 for baselines with a fixed retrieval size (all except PathRAG). CONRAG achieves the highest recall on both benchmarks. Because the document budget is comparable, the improvement does not result from retrieving more. Moreover, retrieval depends on hub suppression. Raising $p_{\text{hub}}$ steadily lifts recall while also increasing retrieval time due to increasing combinatorial complexity of path enumeration. As shown in Figure 4 retrieval time grows strongly after passing $p_{\text{hub}} = 0.8$ while not significantly improving overall document recall which results in a tradeoff between recall and retrieval time for CONRAG.

Table 4: Document Recall (%) on benchmarks (mean ± std over 3 seeds). Best in **bold**, second-best underlined.

| Model | *MuSiQue* | *2WikiMultiHopQA* |
|---|---|---|
| Naïve RAG | 61.1±0.6 | 67.0±0.3 |
| LightRAG | 60.7±0.9 | 69.0±0.1 |
| PathRAG | 62.9±0.8 | 72.2±0.8 |
| HippoRAG 2 | 64.2±2.3 | 79.0±0.3 |
| CONRAG | **70.2±0.5** | **84.1±0.6** |

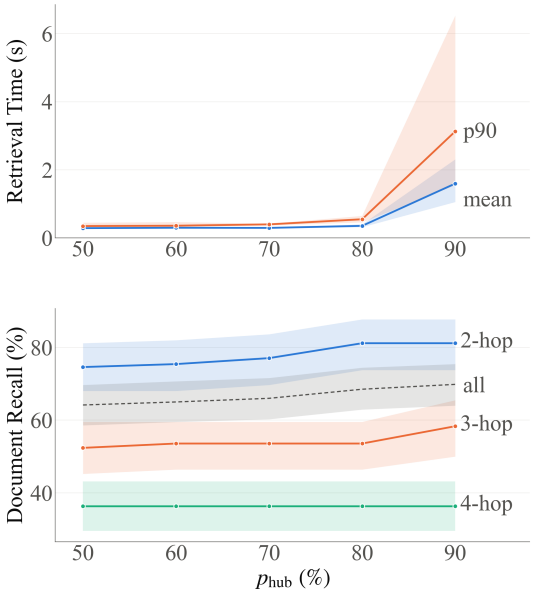


Figure 4: **Hub suppression.** $p_{\text{hub}}$ and retrieval time in seconds and document recall in %. Shaded bands are 95% bootstrap confidence intervals.

## 6 CONCLUSION

In this paper, we introduced CONRAG, a lightweight RAG framework for multi-hop relation inference between two known entities. CONRAG constructs a relation-free entity-document graph using corpus co-occurrence and LLM-based entity filtering, and retrieves paths that explicitly connect the two endpoints through intermediate bridge entities and supporting evidence. We also formalised the task and derived evaluation benchmarks from MuSiQue and 2WikiMultiHopQA by treating annotated reasoning chains as targets between known endpoints. Across both benchmarks, CONRAG improves bridge-entity and reasoning-chain recovery over strong RAG baselines, while also achieving higher document recall and substantially lower graph-indexing token requirements. These results support endpoint-constrained path retrieval as an effective and index-efficient approach for evidence-grounded relation discovery. More broadly, this work motivates greater attention to multi-hop relation inference as a first-class retrieval problem. Recovering *how* known entities are connected is important for scientific discovery, literature analysis, and other knowledge-intensive settings where the connecting evidence matters as much as the endpoint answer. Our results suggest that this capability does not require richly extracted knowledge graphs: A lightweight, relation-free representation combined with endpoint-constrained retrieval already provides a strong and efficient foundation for relation discovery.

**Limitations & Future Work.** Our dataset inherits the gold decompositions of MuSiQue and 2WikiMultiHopQA as the only reference chains. Two entities may, however, be connected through the corpus by several valid routes. Currently, every deviation from the reference is scored as an error, although alternative relations can be valid. Reported CR, CP, and BCM should therefore be read as conservative lower bounds. A second limitation is retrieval time: The cost of path enumeration grows with the number of high-degree entities admitted to the traversal. Hub suppression thus trades connections that run through high-degree entities for tractability. Beyond these limitations, we see the reinterpretation of MHQA datasets as a more general contribution. A benchmark that provides question decompositions and intermediate answers can be reread as a relation inference benchmark. With this, CONRAG opens several directions for future work, including extending the setting to branching questions with multiple anchors, purpose-built relation inference benchmarks, and retrievers trained directly on chain rather than answer supervision.

### AI USE STATEMENT

In this work, we used generative AI tools for tasks requiring mandatory disclosure. As outlined in section 3.5, we used LLMs to reformat the dataset MuSiQue by extracting anchor entities from decomposition questions. Our methodology was developed iteratively, including LLM feedback. To implement CONRAG and baseline models, we used the support of LLMs for writing code. For manuscript editing, we used generative AI to improve clarity and readability, to help identify potentially relevant literature and assess the coverage of related work. All AI-assisted outputs were subsequently reviewed by the authors: Suggested references were manually verified against the original sources, manuscript changes were inspected and approved by the authors, and AI-assisted code was reviewed for correctness. The authors take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the assistance of generative AI.

### REPRODUCIBILITY STATEMENT

To support reproducibility, we document every component needed to rerun our experiments. The full CONRAG pipeline is specified in Section 3, with retrieval summarised in Algorithm 1 and all hyperparameters fixed across datasets and listed in Table 5. Our benchmarks are derived from the public validation splits of MuSiQue and 2WikiMultiHopQA, and Section 3.5 together with Appendix B.3 describes the question selection, anchor and bridge extraction, and corpus pooling in detail. Baselines are taken from their public repositories and configured as described in Appendix B.2, and every prompt used in the pipeline, from benchmark construction to evaluation, is included in Appendix E. Computational requirements are reported in terms of token cost for indexing and generation (Table 3). Code and derived benchmark data are available in an anonymous repository.

## REFERENCES

Kemafor Anyanwu and Amit Sheth. $\rho$-queries: enabling querying for semantic associations on the semantic web. In *Proceedings of the 12th international conference on World Wide Web*, pp. 690–699, 2003.

Sumit Bhatia, Purusharth Dwivedi, and Avneet Kaur. Tell me why is it so? explaining knowledge graph relationships by finding descriptive support passages, 2018. URL https://arxiv.org/abs/1803.06555.

Pere-Lluís Huguet Cabot and Roberto Navigli. Rebel: Relation extraction by end-to-end language generation. In *Findings of the association for computational linguistics: EMNLP 2021*, pp. 2370–2381, 2021.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 30183–30191. AAAI Press, 2026. doi: 10.1609/AAAI.V40I36.40268. URL https://doi.org/10.1609/aaai.v40i36.40268.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. Cognitive graph for multi-hop reading comprehension at scale. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2694–2703, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1259. URL https://aclanthology.org/P19-1259/.

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

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and

Rayid Ghani (eds.), *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 1156–1165. ACM, 2014. doi: 10.1145/2623330.2623677. URL https://doi.org/10.1145/2623330.2623677.

Christos Faloutsos, Kevin S McCurley, and Andrew Tomkins. Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 118–127, 2004.

Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. Minirag: Towards extremely simple retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2501.06713.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on RAG meeting llms: Towards retrieval-augmented large language models. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 6491–6501. ACM, 2024. doi: 10.1145/3637528.3671470. URL https://doi.org/10.1145/3637528.3671470.

Lujun Fang, Anish Das Sarma, Cong Yu, and Philip Bohannon. Rex: explaining relationships between entity pairs, 2011. URL https://arxiv.org/abs/1111.7170.

Bernard A Galler and Michael J Fisher. An improved equivalence algorithm. *Communications of the ACM*, 7(5):301–303, 1964.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2023. URL https://arxiv.org/abs/2312.10997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=hkujvAPVsg.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. More data, more relations, more context and more openness: A review and outlook for relation extraction. In Kam-Fai Wong, Kevin Knight, and Hua Wu (eds.), *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 745–758, Suzhou, China, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.aacl-main.75. URL https://aclanthology.org/2020.aacl-main.75/.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Yuntong Hu, Zhihan Lei, Zhongjie Dai, Allen Zhang, Abhinav Angirekula, Zheng Zhang, and Liang Zhao. CG-RAG: research question answering by citation graph retrieval-augmented llms. In Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne (eds.), *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pp. 678–687. ACM, 2025. doi: 10.1145/3726302.3729920. URL https://doi.org/10.1145/3726302.3729920.

Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. KET-RAG: A cost-efficient multi-granular indexing framework for graph-rag. In Luiza Antonie, Jian Pei, Xiaohui Yu, Flavio Chierichetti, Hady W. Lauw, Yizhou Sun, Srinivasan Parthasarathy, Wee Hyong Tok, Andrew Tomkins, Qi He, Ambuj Singh, and Haixun Wang (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pp. 1003–1012. ACM, 2025. doi: 10.1145/3711896.3737012. URL https://doi.org/10.1145/3711896.3737012.

Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In Regina Barzilay and Mark Johnson (eds.), *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 529–539, Edinburgh, Scotland, UK., 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1049/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Keming Lu, I-Hung Hsu, Wenxuan Zhou, Mingyu Derek Ma, and Muhao Chen. Multi-hop evidence retrieval for cross-document relation extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10336–10351, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.657. URL https://aclanthology.org/2023.findings-acl.657/.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5755–5772, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.395. URL https://aclanthology.org/2022.acl-long.395/.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Anh Tuan Luu. HypergraphRAG: Retrieval-augmented generation via hypergraph-structured knowledge representation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ravS5h8MNg.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. Multi-hop question answering. *Foundations and Trends® in Information Retrieval*, 17(5):457–586, 2024.

Congmin Min, Sahil Bansal, Joyce Pan, Abbas Keshavarzi, Rhea Mathew, and Amar Viswanathan Kannan. Towards practical graphrag: Efficient knowledge graph construction and hybrid retrieval at scale, 2025. URL https://arxiv.org/abs/2507.03226.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle (eds.), *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3866–3878, Santa Fe, New Mexico, USA, 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1326/.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017. doi: 10.1162/tacl_a_00049. URL https://aclanthology.org/Q17-1008/.

Giuseppe Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *International semantic web conference*, pp. 622–639. Springer, 2015.

Mykhailo Poliakov and Nadiya Shvai. Multi-meta-rag: Improving rag for multi-hop queries using database filtering with llm-extracted metadata. In *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*, pp. 334–342. Springer, 2024.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Junyoung Son, Jinsung Kim, Jungwoo Lim, Yoonna Jang, and Heuiseok Lim. Explore the way: Exploring reasoning path by bridging entities for effective cross-document relation extraction. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6755–6761, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.450. URL https://aclanthology.org/2023.findings-emnlp.450/.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Don R Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.

Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM (JACM)*, 22(2):215–225, 1975.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Nikos Voskarides, Edgar Meij, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. Learning to explain entity relationships in knowledge graphs. In Chengqing Zong and Michael Strube (eds.), *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)*, pp. 564–574, Beijing, China, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1055. URL https://aclanthology.org/P15-1055/.

Fengqi Wang, Fei Li, Hao Fei, Jingye Li, Shengqiong Wu, Fangfang Su, Wenxuan Shi, Donghong Ji, and Bo Cai. Entity-centered cross-document relation extraction. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9871–9881, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.671. URL https://aclanthology.org/2022.emnlp-main.671/.

Ning Wang, Kuanyan Zhu, Daniel Yuehwoon Yee, Yitang Gao, Shiying Huang, Zirun Xu, and Sainyam Galhotra. Pruning minimal reasoning graphs for efficient retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2602.04926.

Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 564–573, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1060. URL https://aclanthology.org/D17-1060/.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=EMHoBG0avc1.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 764–777, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1074. URL https://aclanthology.org/P19-1074/.

Yuan Yao, Jiaju Du, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. CodRED: A cross-document relation extraction dataset for acquiring knowledge in the wild. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4452–4472, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.366. URL https://aclanthology.org/2021.emnlp-main.366/.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 35–45, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004. URL https://aclanthology.org/D17-1004/.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39, 2024.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 8912–8924, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.449. URL https://aclanthology.org/2025.naacl-long.449/.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. LinearRAG: Linear graph retrieval augmented generation on large-scale corpora. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mCtfkypdm6.

# A Extended Related Work

**Graph-based RAG variants.** Beyond GraphRAG, LightRAG, and HippoRAG, a number of methods adapt the graph or the retrieval step. HyperGraphRAG (Luo et al., 2025) represents $n$-ary relations as hyperedges, and RAPTOR (Sarthi et al., 2024) replaces the graph with a tree of recursive summaries. KG$^2$RAG (Zhu et al., 2025) expands initially retrieved chunks with graph neighbours. KET-RAG (Huang et al., 2025) pairs a small LLM-extracted skeleton with a keyword-document bipartite graph to reduce indexing cost. All of these still rely on an LLM pass over the corpus for entity and relation extraction or summarisation.

**Cost-efficient graph construction for RAG.** Several more works reduce the LLM cost of building graphs for retrieval. CG-RAG (Hu et al., 2025) avoids extraction entirely by navigating citation links between academic papers. Min et al. (2025) replace the LLM with SpaCy's dependency parser for graph construction, and Wang et al. (2026) use REBEL as a triplet extractor to build multiple smaller triplet graphs that are orchestrated during retrieval. Without any graph, Multi-Meta-RAG (Poliakov & Shvai, 2024) uses extracted metadata to prefilter documents over several steps. ConRAG requires no external structure such as citation links, and no relation extractor at all: The graph is relation-free, its entity nodes are surface mentions mined from cooccurrence statistics and qualified by an LLM. Its retrieval also serves a different goal. ConRAG searches for paths between two known endpoints, which is what makes the intermediate entities and documents available as output.

# B Extended Experimental Setting

## B.1 Additional ConRAG Implementation Details

Table 5: Free parameters of ConRAG and the values used in all experiments.

| Stage | Symbol | Value | Role |
|---|---|---|---|
| Entity qualification | $n_{\max}$ | 5 | maximum phrase length in candidate mining |
| | $B$ | 200 | batch size for containment-aware adjudication |
| Indexing | $\theta$ | 0.8 | embedding similarity threshold for entity node merging |
| Traversal | $k_{\text{seed}}$ | 20 | number of initial vector retrieval seed documents per question |
| | $k_{\text{fallback}}$ | 4 | number of vector retrieval fallback documents for an entity not contained in a document |
| | $p_{\text{hub}}$ | 0.9 | degree quantile above which an entity is suppressed |
| | $L_{\max}$ | 5 | maximum path length in documents |
| Selection | $k_{\text{path}}$ | 5 | paths retained per length |
| | $\mathcal{L}$ | $\{3, 4, 5\}$ | target path lengths |

Table 5 lists the free parameters of ConRAG and the values used in all experiments. Entity qualification is governed by the maximum phrase length $n_{\max}$ and the filtering batch size $B$, indexing by the merging threshold $\theta$, traversal by the seed counts $k_{\text{seed}}$ and $k_{\text{fallback}}$, the hub quantile $p_{\text{hub}}$ and the maximum path length $L_{\max}$, and path selection by $k_{\text{path}}$ and the target lengths $\mathcal{L}$. These values are fixed across both benchmarks rather than tuned per dataset. With $p_{\text{hub}} = 0.9$, entities whose document frequency exceeds the 90th percentile of the degree distribution are excluded from path enumeration.

## B.2 Baselines Implementation

### B.2.1 Naïve RAG

Naïve RAG (Lewis et al., 2020) serves as the non-graph reference point: Single-shot dense retrieval over the same pooled corpus, without query rewriting, reranking or iterative retrieval. Every pooled document is indexed as one unit. documents are embedded with *text-embedding-3-small* using 1,536

dimensions. We store them in an exact FAISS inner-product index. At query time the relation question $q$ is embedded with the same encoder and the top $k = 10$ documents by $\cos(\mathbf{v}_q, \mathbf{v}_d)$ are retrieved. The retrieved documents are inserted as a numbered list of title-text pairs into the shared generation prompt of Appendix E.3. The response is generated by *GPT-4o*. Indexing requires no LLM call, which is why no construction cost is reported in Table 3.

#### B.2.2 LIGHTRAG

LightRAG (Guo et al., 2025) serves as one of the graph-based reference points. Entity and relationship extraction is performed with *GPT-4o-mini*. Entities, relationships and chunks are embedded with *text-embedding-3-small* using 1,536 dimensions and stored in LightRAG's native NanoVectorDB and NetworkX backends. Queries are run in hybrid mode, with retrieval capped at $k = 10$ documents. The retrieved context is inserted into the shared generation prompt and the response is generated by *GPT-4o*.

#### B.2.3 PATHRAG

PathRAG (Chen et al., 2026) is the baseline whose retrieval mechanism is closest to CONRAG's own. Since PathRAG and LightRAG share their underlying graph (Chen et al., 2026), we use the same NanoVectorDB index and NetworkX graph for MuSiQue and 2WikiMultiHopQA which explains the equal construction cost in Table 3. Vector search retrieves the $T$=10 entity nodes and relationship edges most similar to these keywords, forming the candidate pool $V_q$. This pool is then pruned to the $K$=15 highest-scoring relational paths. The per-hop decay rate is set at $\alpha$=0.8 and the decision retention threshold is set at 0.3. The response is generated by *GPT-4o*.

#### B.2.4 HIPPORAG 2

HippoRAG 2 (Gutiérrez et al., 2025) serves as the graph-based RAG reference point designed for multi-hop reasoning. *GPT-4o-mini* performs entity and relation extraction. *text-embedding-3-small* using 1,536 dimensions embeds both documents and extracted facts. Synonymy edges are added between entity nodes whose embeddings exceed a cosine similarity threshold of 0.8. We retrieve the top $k = 10$ documents via PPR with a damping factor of 0.5. We use *GPT-4o* as recognition memory and as generator and use the shared generation prompt with the retrieved documents.

### B.3 DATASETS

#### B.3.1 ORIGINAL DATASETS

**MuSiQue.** MuSiQue (Trivedi et al., 2022) is built from single-hop questions of five Wikipedia-based reading comprehension datasets. Two questions are composed if the answer of the first is a named entity mentioned in the second. A composition is kept only if no subquestion can be answered without its predecessor's answer. Each question is described by the graph of its subquestions, which has two to four hops and one of six shapes. In the three linear shapes every subquestion depends only on the answer of the one before it. The question names a single starting entity, and the intermediate answers form one ordered chain to the final answer: *What currency is used where Billy Giles died?* resolves *Billy Giles* → *Belfast* → *Northern Ireland* → *pound sterling*. In the three branching shapes a subquestion depends on the answers of two predecessors that are independent of each other. *When was the first establishment that McDonaldization is named after opened in the country Horndean is located?* separately resolves *McDonaldization* → *McDonald's* and *Horndean* → *England*, and only their combination (*When did the first McDonald's open in England?*) yields the answer *1974*. Such a question starts from two entities, and its intermediate answers are not connected to each other, so they form two converging paths rather than one chain between an anchor and a target. Every question ships with its decomposition containing subquestions, intermediate answers, one supporting paragraph per hop and a context of 20 paragraphs including hard distractors. The validation split answerable questions has 2,417 questions containing 1,252 two-hop, 760 three-hop and 405 four-hop questions.

**2WikiMultiHopQA.** 2WikiMultiHopQA (Ho et al., 2020) is generated from templates over Wikidata triples that are aligned with the summary paragraphs of the corresponding Wikipedia articles.

It has four question types: *comparison*, *inference*, *compositional* and *bridge comparison*. Compositional and inference questions are both built from two chained triples $(e, r_1, e_1)$ and $(e_1, r_2, e_2)$, where the paragraph of $e$ must mention the bridge $e_1$ and the paragraph of $e_1$ must mention the answer $e_2$. Inference questions additionally collapse $r_1$ and $r_2$ into a single relation through one of 28 logical rules (e.g. *father* $\wedge$ *father* $\Rightarrow$ *paternal grandfather*), so the question does not hint at the bridge. Each question is annotated with its evidence triples and a context of 10 paragraphs, the gold paragraphs plus tf-idf retrieved distractors. The dataset has 192,606 questions, where 12,576 questions belong to the validation set.

Both benchmarks thus annotate the intermediate entities of every question explicitly, which is what allows them to be reread as relation inference benchmarks in Section B.3.2.

### B.3.2 Processing for multi-hop relation inference

**Scope.** Of the 2,417 MuSiQue validation questions we keep the linear compositions *2hop*, *3hop1* and *4hop1* and drop the branching types *3hop2*, *4hop2* and *4hop3*, which start from two independent anchors. This results in 2066 questions. Of the 12,576 2WikiMultiHopQA questions we keep *compositional* and *inference* resulting in 6785 questions. *comparison* and *bridge comparison* compare two anchors and have no bridge entity.

**Anchor, target, bridges and chain.** The target $b$ is the answer of the source question, the anchor $a$ the entity its reasoning chain starts from. In 2WikiMultiHopQA every question carries a list of (subject, relation, object) evidence triples: The subject of the first triple is the anchor, the objects of all but the last triple are the bridges, and the triples in order give the gold links. In MuSiQue the decomposition lists one subquestion per hop. If the first subquestion has the structured form `<entity> » <relation>` the anchor is parsed directly. If the decomposition occurs to be a subquestion it is extracted by *GPT-4o-mini* by using the prompt in Appendix E.1. Intermediate answers are the bridges. Anchor, intermediate answers and target chained in decomposition order form the gold links. Each anchor-target pair is inserted into one of six paraphrases of *How are {anchor} and {target} related?*, drawn uniformly at random.

**Anchor extraction quality.** LLM extraction is the only non-deterministic step. The extracted anchor is a verbatim substring of the source question for 92.5% of the affected questions and occurs verbatim in the supporting document of the first decomposition step for 77.4%. In a manual check of 100 random extractions, 92 named the entity the gold chain starts from.

**Worked example: 2WikiMultiHopQA [compositional]**

**Source:** *Where was the director of film Whispers (1990 Film) born?*
**Evidence:**
[Whispers - director - Douglas Jackson]
[Douglas Jackson - place of birth - Montreal, Quebec]

---

**Anchor** *Whispers*, **Target** *Montreal, Quebec*, **Bridge** {Douglas Jackson}.
**Links:**
[`Whispers -> Douglas Jackson`]
[`Douglas Jackson -> Montreal, Quebec`]
**Relation question:** *How are Whispers and Montreal, Quebec related?*

**Worked example: MuSiQue [4-hop]**

**Source:** *The German priest, who wanted Jean Hengen's religious denomination to reform, preached a sermon on Marian devotion a month before his death in what German state?*
**Decomposition:**
(1) `Jean Hengen » religion` ⇒ Catholic Church
(2) *who wanted #1 to reform* ⇒ Martin Luther
(3) *Where did #2 preach a sermon on Marian devotion a month before his death?* ⇒ Wittenberg
(4) `#3 » located in the administrative territorial entity` ⇒ Saxony-Anhalt.

---

**Anchor** *Jean Hengen*, **Target** *Saxony-Anhalt*, **Bridges** {Catholic Church, Martin Luther, Wittenberg}.
**Links:**
[`Jean Hengen -> Catholic Church`]

```
[Catholic Church -> Martin Luther]
[Martin Luther -> Wittenberg]
[Wittenberg -> Saxony-Anhalt]
```
**Relation question:** *What links Jean Hengen to Saxony-Anhalt?*

Chain recall and precision are computed against these link strings after lower-casing and accent folding, bridge contains match against the bridge strings.

#### B.3.3 QUESTION STATISTICS

An $n$-hop source question yields $n$ gold links, $n-1$ bridges and $n$ gold documents. MuSiQue contains 1,252 with two, 568 with three and 246 questions with four hops. All 6,785 2WikiMultiHopQA questions are 2-hop with a single bridge. For each dataset three samples of 1,000 questions are drawn with seeds 123, 456, and 789.

#### B.3.4 POOLED CORPORA

The retrieval corpus of each dataset pools the context paragraphs of all validation questions and deduplicates on whitespace collapsed title and text. A MuSiQue question originally comes with 20 context paragraphs of which two to four are supporting. A 2WikiMultiHopQA question with 10 of which two are supporting. After pooling every question is answered against the full corpus rather than its own paragraph bundle. The full corpus for MuSiQue contains approximately 26.3k documents and 56.7k for 2WikiMultiHopQA.

#### B.3.5 LIMITATIONS OF THE DERIVED BENCHMARK

Table 6: CONRAG on full vs clean question subsets. Values are mean ± SD across seeds.

| Dataset | Subset | Avg. $\lvert\mathcal{Q}\rvert$ | BCM | CR | CP |
|---|---|---|---|---|---|
| MuSiQue | full | 1000.0 | 71.7 ± 1.3 | 28.4 ± 1.3 | 34.5 ± 2.4 |
| MuSiQue | clean | 840.3 | 74.8 ± 0.3 | 32.4 ± 0.7 | 39.2 ± 0.4 |
| 2WikiMultiHopQA | full | 1000.0 | 81.0 ± 1.4 | 49.4 ± 1.3 | 48.1 ± 0.9 |
| 2WikiMultiHopQA | clean | 985.3 | 81.1 ± 1.3 | 49.5 ± 1.3 | 48.2 ± 1.2 |

Gold chains are only as complete as the source decompositions. A valid alternative path between anchor and target is scored as a false link. Moreover, not every gold link is supported by the document text. We call an entity grounded in a document if all of its tokens occur in that document after lower-casing and accent folding, which is the most tolerant matching rule the retriever itself applies. On MuSiQue the anchor is not grounded in any gold document for 163 of the 2,066 questions. The target is not grounded for 6 questions. A bridge is not grounded in both documents it connects for 189 of the 3,126 MuSiQue bridges affecting 186 questions and for 95 of the 6,785 2WikiMultiHopQA bridges. Taken together, 312 MuSiQue questions and 95 2WikiMultiHopQA questions contain at least one gold link that no document supports. As shown in Table 6 restricting the evaluation to the 1,754 grounded MuSiQue questions raises every metric for CONRAG.

## C COMPREHENSIVE MULTI-HOP RELATION INFERENCE PERFORMANCE

### C.1 DETAILED RESULTS OVERVIEW

Table 7 stratifies the results of Table 1 by hop count. CONRAG leads BCM in every block, by 11.9, 13.8 and 4.9 points on two-, three- and four-hop MuSiQue questions, with a relative decay comparable to the baselines. The chain metrics separate the systems clearly on two-hop questions, where CONRAG gains points in chain precision and chain recall on 2WikiMultiHopQA and outperforms HippoRAG 2 on MuSiQue in chain recall. At three and four hops every system recovers fewer than one in ten and one in twenty gold links. The the nominal lead of LightRAG at two hops on precision, lies within one standard deviation. At four hops CONRAG's chain precision lies below baselines. Notably, CONRAG still names most of the gold bridges on four-hop questions and leads in chain recall.

Table 7: Hop-stratified relation retrieval performance across systems. BCM = Bridge Contains Match Accuracy, CR = chain recall, and CP = chain precision. Values are mean ± standard deviation across available seeds. Bold indicates the best score within each dataset-hop block and metric.

| *Dataset* | *Hops* | *Model* | *BCM* | *CR* | *CP* |
|---|---|---|---|---|---|
| 2WikiMultiHopQA | 2 | ConRAG | **81.0% ± 1.4** | **49.4% ± 1.2** | **48.1% ± 0.9** |
| | | HippoRAG 2 | 67.0% ± 1.0 | 40.2% ± 0.9 | 34.9% ± 2.1 |
| | | LightRAG | 72.1% ± 0.2 | 40.2% ± 1.0 | 35.7% ± 1.0 |
| | | PathRAG | 64.9% ± 0.2 | 30.4% ± 1.5 | 35.8% ± 1.8 |
| | | Naïve RAG | 61.6% ± 1.5 | 33.4% ± 2.4 | 40.1% ± 2.3 |
| MuSiQue | 2 | ConRAG | **84.2% ± 1.2** | **43.7% ± 0.9** | 38.5% ± 1.5 |
| | | HippoRAG 2 | 72.3% ± 2.1 | 35.2% ± 0.7 | **40.2% ± 2.0** |
| | | LightRAG | 72.1% ± 2.6 | 30.3% ± 0.3 | 31.1% ± 0.5 |
| | | PathRAG | 71.9% ± 1.6 | 26.7% ± 0.3 | 36.9% ± 1.4 |
| | | Naïve RAG | 72.2% ± 0.8 | 29.8% ± 0.6 | 30.9% ± 0.8 |
| | 3 | ConRAG | **57.6% ± 0.3** | **9.4% ± 0.2** | **25.5% ± 0.5** |
| | | HippoRAG 2 | 43.8% ± 1.5 | 5.9% ± 0.6 | 24.9% ± 1.7 |
| | | LightRAG | 40.1% ± 2.7 | 5.2% ± 0.9 | 19.0% ± 2.1 |
| | | PathRAG | 40.8% ± 1.0 | 3.0% ± 0.2 | 19.4% ± 2.0 |
| | | Naïve RAG | 42.8% ± 0.1 | 4.3% ± 0.7 | 16.6% ± 3.1 |
| | 4 | ConRAG | **36.8% ± 2.6** | **3.8% ± 0.5** | 12.0% ± 0.8 |
| | | HippoRAG 2 | 31.8% ± 4.1 | 3.2% ± 0.7 | **17.1% ± 1.8** |
| | | LightRAG | 28.4% ± 2.7 | 3.7% ± 0.8 | 16.5% ± 2.7 |
| | | PathRAG | 30.6% ± 2.7 | 2.0% ± 1.0 | 15.5% ± 4.5 |
| | | Naïve RAG | 31.9% ± 2.6 | 3.1% ± 0.5 | 12.9% ± 1.6 |

### C.2 Win Case Analysis

To illustrate how ConRAG's path enumeration translates into the gains reported above, we contrast its output with that of HippoRAG 2, the strongest baseline on chain recall, on a two-hop MuSiQue question. The example is representative of a win pattern in our runs. Both systems retrieve the anchor document, but only ConRAG reaches the bridge document, since the shared entity *Palembang* links the anchor document to the target document in the entity-document graph and the resulting path is scored highly by query support. HippoRAG 2 instead spreads activation from the query entities toward semantically related wartime documents that mention neither endpoint jointly. As the bridge document is the only document that ties the birthplace of the anchor to the date of the target, its presence in the context decides whether a chain can be emitted at all.

**Win case: What links Marzuki Alie to 15 February 1942?**

**Gold chain:** Marzuki Alie → Palembang → 15 February 1942.
**Gold documents:** ***Battle of Palembang***, ***Marzuki Alie***.
**Bridge entity:** Palembang, described in ***Battle of Palembang***. Gold documents are set in bold wherever they appear below.

**ConRAG.** Retrieved documents: ***Battle of Palembang***, ***Marzuki Alie***, *Tengku Tengah Zaharah Mosque*, *Choo Hoey*, *Arie Supriyatna*, *1979 Garuda Fokker F28 crash*, *Mohd Marzuki Yusof*, *Sang Nila Utama*, *Denny Januar Ali*, *Muhammad Shobran*, *Southeast Asia*, *Pacific War*.
**Predicted chain:** Marzuki Alie → Palembang → 15 February 1942.

> *Marzuki Alie was born in Palembang, South Sumatra, on 6 November 1955. Palembang is also the location of the Battle of Palembang, which occurred during the Pacific theatre of World War II from 13–15 February 1942. The connection between Marzuki Alie and 15 February 1942 is through the shared location of Palembang, which ties Marzuki Alie's birthplace to the historical event that took place there on that date.*

**HippoRAG 2.** Retrieved documents (top ten, the documents its prompt carries): ***Marzuki Alie***, *Julius Ellsberry*, *Battle of the Tenaru*, *Abdul Ali Mazari*, *Military history of the United States during World War II*, *September 11 attacks*, *Abu Yahya al-Libi*, *Larry Alcala*, *United Defense M42*, *Marshall Islands*.
**Predicted chain:** none returned.

> *The provided documents do not establish any meaningful connection between Marzuki Alie and the date 15 February 1942. Marzuki Alie is described as a political figure from Indonesia, serving as the speaker of the People's Representative Council and a member of the Partai Demokrat. The date 15 February 1942 is not mentioned in the context, nor is there any event or entity linked to Marzuki Alie that connects him to this specific date. The documents retrieved do not support any relationship beyond coincidental term overlap.*

The bridge document *Battle of Palembang* reaches CONRAG's context but not HippoRAG 2's, whose ten documents are drawn instead from topically adjacent wartime material. The failure is in retrieval rather than generation: The connecting fact is absent from the context, so no amount of reasoning over it recovers the chain.

### C.3 FALSE NEGATIVE ANALYSIS

Table 8: Construction of the false-negative pool (1000 questions per system). **Match** = at least one gold link reproduced exactly; **Abst.** = no chain emitted; **Deny** = a chain was emitted but the prose denies a connection; **Near** = a gold link recovered under a different surface form; **Unsp.** = the predicted chain does not span anchor to target. **Alt** is the remaining pool of alternative-path candidates.

| Dataset | System | Match | Abst. | Deny | Near | Unsp. | Alt |
|---|---|---|---|---|---|---|---|
| *MuSiQue* | CONRAG | 380 | 403 | 25 | 52 | 13 | 127 |
| | HippoRAG 2 | 300 | 511 | 13 | 59 | 10 | 107 |
| | LightRAG | 276 | 548 | 18 | 36 | 26 | 96 |
| | Naïve RAG | 253 | 549 | 26 | 65 | 2 | 105 |
| | PathRAG | 238 | 621 | 11 | 30 | 25 | 75 |
| *2WikiMultiHopQA* | CONRAG | 643 | 173 | 13 | 60 | 4 | 107 |
| | HippoRAG 2 | 489 | 301 | 37 | 52 | 19 | 102 |
| | LightRAG | 496 | 241 | 17 | 96 | 14 | 136 |
| | Naïve RAG | 369 | 417 | 36 | 49 | 12 | 117 |
| | PathRAG | 353 | 397 | 20 | 78 | 13 | 139 |

Table 9: Classification of the alternative-path candidates and blind validation. **Alias**: the chain reaches the gold bridge under a different surface form. **Short**: the chain bypasses the gold bridge and links the endpoints directly. **Degen**: an intermediate node is a description or a restatement of the anchor or target rather than a distinct entity. **Distinct** cases are judged blind against that system's own retrieved documents; a case is **Valid** only if every link is supported, the chain connects anchor to target, and every step is substantive.

| | | | Excluded | | | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | System | Alt | Alias | Short | Degen | Distinct | Valid | Rate |
| *MuSiQue* | CONRAG | 127 | 24 | 5 | 18 | 80 | 70 | 87.5% |
| | HippoRAG 2 | 107 | 20 | 8 | 16 | 63 | 58 | 92.1% |
| | LightRAG | 96 | 18 | 8 | 11 | 59 | 40 | 67.8% |
| | Naïve RAG | 105 | 24 | 6 | 11 | 64 | 55 | 85.9% |
| | PathRAG | 75 | 16 | 7 | 11 | 41 | 20 | 48.8% |
| *2WikiMultiHopQA* | CONRAG | 107 | 19 | 10 | 17 | 61 | 58 | 95.1% |
| | HippoRAG 2 | 102 | 15 | 4 | 18 | 65 | 61 | 93.8% |
| | LightRAG | 136 | 45 | 8 | 15 | 68 | 61 | 89.7% |
| | Naïve RAG | 117 | 29 | 12 | 19 | 57 | 50 | 87.7% |
| | PathRAG | 139 | 44 | 6 | 16 | 73 | 48 | 65.8% |

Since the evaluation protocol accepts only the gold decomposition as reference, a chain that connects anchor and target through a different but valid route is scored as an error. To estimate how much of the reported error is of this kind, we analyse the questions of the first seed from the MuSiQue and 2WikiMultiHopQA relation samples on which a system reproduces no gold link. Table 8 partitions

these questions into abstentions, denials of a connection, near matches under a different surface form, chains that do not span both endpoints and the remaining pool of alternative-path candidates. Table 9 further removes candidates that reach the gold bridge under an alias, bypass it or route through a degenerate node, and validates the distinct remainder blind against the retrieved documents of the respective system. Two findings follow. First, abstentions dominate the false negatives of every system, so the primary failure mode is missing evidence rather than a wrong chain. Second, valid alternative chains are common but do not change the ranking. For CONRAG, 70 of 80 distinct candidates on MuSiQue and 58 of 61 on 2WikiMultiHopQA are supported by its own context. The baselines gain similar absolute amounts, while LightRAG and PathRAG show markedly lower validity rates, indicating that their deviating chains are more often unsupported by the retrieved documents than genuinely alternative. The two cases below illustrate valid alternatives that the metric rejects.

**Case 1: What links Australia to Lyons' death in 1939?**

**Gold path:** Australia → Robert Menzies → Lyons' death in 1939.
**Alternative path:** Australia → Joseph Lyons → Lyons' death in 1939.

**CONRAG** explains the connection as follows:

> *Joseph Lyons was a prominent political figure in Australia, serving as the leader of the United Australia Party (UAP) and as Prime Minister. His death in 1939 marked a significant event in Australian political history, as it led to Robert Menzies assuming the Prime Ministership. Thus, the connection between Australia and Lyons' death in 1939 is through his role as a key political leader in the country.*

The gold chain routes through Robert Menzies, who succeeded Lyons. The alternative routes through Joseph Lyons himself, the person whose death the question names. It is the more direct reading, and the metric scores it zero because neither link string matches.

**Case 2: Explain the relationship between Mickey Mouse and Walt Disney.**

**Gold path:** Mickey Mouse → Minnie Mouse → Walt Disney.
**Alternative path:** Mickey Mouse → Ub Iwerks → Walt Disney.

**CONRAG** explains the connection as follows:

> *Collaboration with Ub Iwerks: Ub Iwerks, a close collaborator of Walt Disney, was instrumental in designing Mickey Mouse. Iwerks created the initial sketches of the character, inspired by earlier ideas and sketches of mice around a photograph of Walt Disney. This collaboration ties Mickey Mouse to Walt Disney through their shared creative process.*

The gold chain routes through Minnie Mouse, a second fictional character. The alternative routes through Ub Iwerks, who drew the original sketches of Mickey Mouse and co-created the character with Disney.

# D MINING AND RETRIEVAL STUDIES

## D.1 ENTITY FILTERING

Figure 5 traces the entity vocabulary through indexing and reports, after each step, the share of surviving candidates together with the share of gold anchors, targets and bridges that still have a node. The document frequency cut already removes about 85 % of the mined candidates. LLM filtering removes roughly two thirds of the remainder, leaving about 4 % and 6 % of the initial vocabulary on MuSiQue and 2WikiMultiHopQA. Bridge coverage, which is what path enumeration depends on, is largely unaffected by these two steps. Anchors and targets lose more, but they are recovered by the endpoint constraint during seeding and need not be present in the graph.

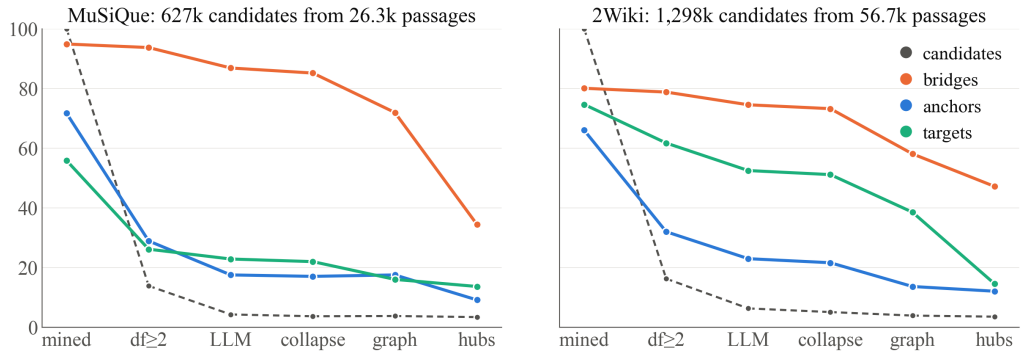


Figure 5: **Entity vocabulary funnel.** Share (%) of mined candidates (dashed) and of gold bridges, targets and anchors that retain an entity node (solid) after each indexing stage on MuSiQue (left) and 2WikiMultiHopQA (right). The *hubs* stage applies only to the traversal adjacency.

# E Overview of LLM Prompts

## E.1 Dataset Anchor Entity Extraction

```
You extract the one NAMED ENTITY that a multi-hop question starts from.

Multi-hop questions nest descriptions inside each other. Somewhere at the bottom of that
nesting sits a single proper name - a person, place, work, organisation or product. That
name is the anchor. Everything wrapped around it is description, not the answer.

"Who is the spouse of the Green performer?" -> Green performer
"Who gives out the prize named after the author of The Wealth of Nations?" -> The Wealth
of Nations
"When was the death penalty abolished in the country near the country where the writer
of The Book Thief is a citizen of?" -> The Book Thief
"Who is the creator of Mickey Mouse's spouse?" -> Mickey Mouse
"What county shares a border with the county where She Did It's performer was born?" ->
She Did It

CRITICAL: return a NAME, never a description.
Never return anything containing "of the", "where", "which", "that", "named after", "the
country", "the person", "the city". Those are descriptions of an entity, not the entity.
If your answer needs a relative clause to make sense, you have the wrong span - go
deeper into the nesting until you reach the bare proper name.

Rules:
1. Copy the name exactly as it appears, with all of its words and any leading "The" that
is part of the title: "The Wealth of Nations", not "Wealth".
2. Almost always 1-4 words. More than that is a sign you returned a description.
3. Never return the thing being asked for, a property ("spouse", "founder"), or a bare
category ("the company", "the film").
4. If several names appear, return the one the chain begins at - the most deeply nested.

Return ONLY a JSON array, one object per question, in order: [{"index": 0, "anchor":
"..."}]
```

This prompt is used once, during benchmark construction (Section 3.5), and is not part of ConRAG itself. It applies only to MuSiQue questions whose first decomposition step is a natural-language subquestion rather than the structured form `<entity> » <relation>`, from which the anchor can be parsed directly. *GPT-4o-mini* receives a batch of such subquestions and returns the single proper name the reasoning chain starts from for each. The few-shot examples and the rules steer the model towards the nested name and away from descriptive spans. The quality of the extracted anchors is assessed in Appendix B.3.2.

### E.2 ENTITY FILTERING

```
You decide which words or short phrases are REAL NAMED ENTITIES.

A true entity should be a concrete value you could assign to an object slot, for
example:
- person: "Emmanuel Macron"
- place: "Paris"
- organisation: "UNESCO"
- event/work/product: "World War II", "The Beatles"
- date/time value: "11 July", "January 19, 2018"
- language name as an object value: "French language", "Swahili"

Mark false if the item is generic, descriptive, grammatical, or not a specific referent.

Examples:
- true: "Paris", "Mozart", "Boeing", "United States", "World War II", "January 19, 2018"
- false: "main", "best", "series", "television", "river", "company", "building",
         "quickly", "running", "theory", "board of directors", "west of the river"

Input items (JSON array with index/text/capitalization_pct):
__ITEMS_JSON__

Return ONLY a JSON array with one object per input item, in order:
[{"index": 0, "entity": true}, {"index": 1, "entity": false}]
```

This prompt implements the LLM-based entity filtering step of Section 3.2 and accounts for the entire LLM token cost of CONRAG's indexing reported in Table 3. *GPT-4o* receives a containment-aware batch of up to $B$ mined candidates as a JSON array that replaces the placeholder. Each item carries its index, its surface string and its capitalisation rate across the corpus, which serves as a signal for distinguishing proper names from common words. The model returns one binary label per candidate. Only candidates labelled with *true* enter the entity vocabulary.

### E.3 GENERATION

```
You describe how two entities are connected, using passages retrieved from a document
graph.
You are given a set of PASSAGES (unique seed and path-selected documents) in this system
prompt, and a QUESTION naming two entities in the user message. Use only the provided
passages as evidence.
Your job is to give the reader the fullest possible account of how these two entities
relate, **as far as the passages support it**. Not a verdict - an explanation.
Relationships can be:
- Example 1
  - Relationship question: "What is the connection between Mickey's Safari in Letterland
  and Walt Disney?"
  - Output-style chain:
    - `"Mickey's Safari in Letterland -> Mickey Mouse"`
    - `"Mickey Mouse -> Walt Disney"`
  - Explanation example:
    - "`Mickey's Safari in Letterland` is part of the `Mickey Mouse` series. `Mickey
    Mouse` was developed by `Walt Disney`.
- Example 2
  - Relationship question: "What links Zubly Cemetery to Richland County?"
  - Output-style chain:
    - `"Zubly Cemetery -> South Carolina"`
    - `"South Carolina -> Columbia"`
    - `"Columbia -> Forest Acres"`
    - `"Forest Acres -> Richland County"`
  - Explanation example:
    - "`Zubly Cemetery` is located in `South Carolina`, which has the capital
    `Columbia`. `Columbia` borders `Forest Acres`, which is located in `Richland
    County`."
- Example 3
  - Relationship question: "What links Houston Kid to Warner Music Group?"
```

```
  - Output-style chain:
    - `"Houston Kid -> Rodney Crowell"`
    - `"Rodney Crowell -> Warner Bros. Records"`
    - `"Warner Bros. Records -> Warner Music Group"`
  - Explanation example:
    - "`Houston Kid` is performed by `Rodney Crowell` whose record label is `Warner
    Bros. Records`. `Warner Bros. Records` belongs to `Warner Music Group`."
## Be comprehensive
Trace every thread the passages offer, not just the first or strongest:
All three are worth describing. Report each connection separately, strongest first, and
say which kind it is so the reader can weigh it. A single entity pair often has several
distinct threads running through different intermediates - find them all.
## Be diverse
Cover different *kinds* of connection, not several restatements of one. Prefer breadth
across relation types over depth on one.
## Be empowering
Write so the reader can understand the connection and judge it themselves:
- Name the intermediates. "Connected through their producer" is weak; "connected through
Steve Hillage, who performed on the album and is married to Miquette Giraudy" is useful.
- Give the specifics the passages contain - dates, roles, places, titles.
- Where the passages leave a gap, name the gap. Saying what is missing is part of a
useful explanation.
## Stay grounded
Everything must come from the passages. Do not use outside knowledge, and do not invent
a link to make a tidier story.
**A shared word is not itself a relationship.** Two passages both containing "1996" or
"london" tells you they share a term, nothing more.
If the passages genuinely show nothing beyond coincidental term overlap, say so and
describe what they do show - but look properly first.
In every case, only assert what the retrieved passages explicitly support.
Moreover, you shall explicitly chain related entities that relate to each other directly
in the format outlined in the output
## Output
Return ONLY a JSON object:
{
  "chains": [
    [
      "Entity A -> Bridge 1",
      "Bridge 1 -> Bridge 2",
      "Bridge 2 -> Entity B"
    ]
  ],
  "explanation": "a comprehensive prose explanation of how the entities are related (or
  not), grounded only in the provided passages"
}
Each chain is a list of directed entity links in the form:
`"from_entity -> to_entity"`.
`chains` must contain one or more chains supported by the passages.
If no meaningful relationship is supported, return `"chains": []` and explain that only
incidental overlap (or no overlap) was found.
Do not output any other keys besides `chains` and `explanation`.
```

All systems, CONRAG and the four baselines, share this generation prompt with *GPT-4o*. Only the passage block differs and holds the context retrieved by the respective method, so differences in the results stem from retrieval rather than from prompt design. The retrieved passages are placed in the system prompt and the relation question in the user message. The model returns a JSON object with two fields that mirror the output $\mathcal{O}(q) = (\mathcal{C}, y)$ of Eq. 1. The grounding instructions allow the model to return an empty chain list when the passages support no connection.

### E.4 EVALUATION

```
---Role---
You are an expert tasked with evaluating two answers to the same question based on four
criteria: \
**Comprehensiveness**, **Diversity**, and **Empowerment**.
```

```
---Goal---
You will evaluate two answers to the same question based on four criteria:
**Comprehensiveness**, \
**Diversity**, and **Empowerment**.
- **Comprehensiveness**: How much detail does the answer provide to cover all aspects
and details of the question?
- **Diversity**: How varied and rich is the answer in providing different perspectives
and insights on the question?
- **Empowerment**: How well does the answer help the reader understand and make informed
judgments about the topic?
For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain
why. Then, select an \
overall winner based on these three categories.
Here is the question: {question}
Here are the two answers:
**Answer 1:**
{answer1}
**Answer 2:**
{answer2}
Evaluate both answers using the three criteria listed above and provide detailed
explanations for each criterion.
Output your evaluation in the following JSON format:
{{
    "Comprehensiveness": {{
        "Winner": "[Answer 1 or Answer 2]",
        "Explanation": "[Provide explanation here]"
    }},
    "Diversity": {{
        "Winner": "[Answer 1 or Answer 2]",
        "Explanation": "[Provide explanation here]"
    }},
    "Empowerment": {{
        "Winner": "[Answer 1 or Answer 2]",
        "Explanation": "[Provide explanation here]"
    }},
    "Overall Winner": {{
        "Winner": "[Answer 1 or Answer 2]",
        "Explanation": "[Summarize why this answer is the overall winner based on the
        three criteria]"
    }}
}}
```

For the qualitative comparison in Table 2 we adopt the pairwise judging prompt of LightRAG (Guo et al., 2025). For each relation question, *GPT-4o-mini* is shown the explanation generated by CONRAG and that of one baseline as Answer 1 and Answer 2. It selects a winner for comprehensiveness, diversity and empowerment, justifies each choice, and then names an overall winner based on the three criteria. Win rates are the share of questions on which CONRAG is selected.