CoRAG: Enhancing Hybrid Retrieval-Augmented Generation through a Cooperative Retriever Architecture

Zaiyi Zheng[1], Song Wang[1], Zihan Chen[1], Yaochen Zhu[1], Yinhan He[1], Liangjie Hong[2], Qi Guo[2], Jundong Li[1]

[1]University of Virginia, [2]LinkedIn

{sjc4fq,sw3wv,brf3rx,uqp4qh,nee7ne,jl6qk}@virginia.edu,

{qguo,liahong}@linkedin.com

Abstract

Retrieval-Augmented Generation (RAG) is introduced to enhance Large Language Models (LLMs) by integrating external knowledge. However, conventional RAG approaches treat retrieved documents as independent units, often overlooking their interdependencies. Hybrid-RAG, a recently proposed paradigm that combines textual documents and graph-structured relational information for RAG, mitigates this limitation by collecting entity documents during graph traversal. However, existing methods only retrieve related documents from local neighbors or subgraphs in the knowledge base, which often miss relevant information located further away from a global view. To overcome the above challenges, we propose CoRAG that dynamically chooses whether to retrieve information through direct textual search or explore graph structures in the knowledge base.[1] Our architecture blends different retrieval results, ensuring the potentially correct answer is chosen based on the query context. The textual retrieval components also enable global retrieval by scoring non-neighboring entity documents based on semantic relevance, bypassing the locality constraints of graph traversal. Experiments on semi-structured (relational and textual) knowledge base QA benchmarks demonstrate the outstanding performance of CoRAG.

1 Introduction

Recent advances in Large Language Models (LLMs) (Chen et al., 2025) have led to impressive achievements across diverse tasks such as question answering (Daull et al., 2023) and drug discovery (Zheng et al., 2024). Despite their capabilities, LLMs remain prone to factual errors and hallucinations (e.g., answering "*The capital of Canada is Toronto*"). To mitigate this, Retrieval-Augmented Generation (RAG) (Gao et al., 2024) enhances LLMs by retrieving relevant knowledge from external data sources (Zhu et al., 2025). However, traditional RAG treats retrieved documents independently, ignoring their interdependencies (Wang et al., 2023). Graph-RAG (Peng et al., 2024) addresses this by leveraging graph-structured knowledge bases (Yih et al., 2016) to model relationships among retrieved entities, enabling richer context and deeper semantics for improved generation (Peng et al., 2024).

Most Graph-RAG methods focus on graph traversal over knowledge graphs (Schmelzeisen et al., 2021) to retrieve relevant graph components, such as nodes (Li et al., 2024d), triplets (Li et al., 2024c), and paths (Lo and Lim, 2023), based on the relationships between entities in the graph. Despite their strong performance in knowledge base question answering (KBQA) (Diefenbach et al., 2018), they still heavily rely on graph structures of the knowledge base and lack consideration of textual information. Such a paradigm restricts their effectiveness in handling questions that require a higher level of textual comprehension (Lee et al., 2024). Specifically, KBs only represent knowledge as discrete triplets (e.g., (high-altitude residents, prone to, oxygen deficiency)), which lack the contextual understanding needed to answer analytical questions like "*Why are people in high-altitude regions more prone to oxygen deficiency?*"

Recently, there has been increasing attention on using semi-structured knowledge bases (SKB) (Wu et al., 2024b) to complement KBQA with supplementary textual information, where each entity is associated with a document. This has led to the development of Hybrid-RAG (Yuan et al., 2024), a paradigm that integrates relational and textual information for retrieval within the RAG framework. As shown in Figure 1, we retrieve entities either from the entire set of documents based on textual relevance (`Textual`), or from graph neighbours centered on topic entities in the query (`Relational`).

[1]https://github.com/zhengzaiyi/CoRAG

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 16088–16101
November 4-9, 2025

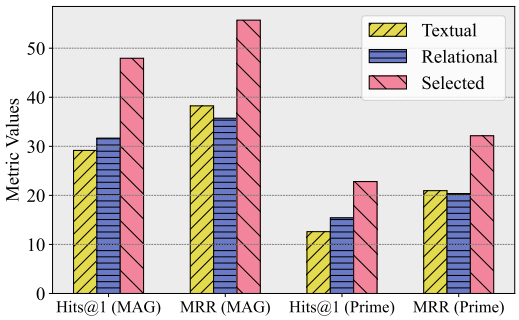

Figure 1: Base (Textual and Relational) and hybrid (Selected) retriever performance analysis on the MAG and Prime datasets. Both types of base retrievers demonstrate a large performance gap from the theoretical upper bound (Selected), indicating significant potential for improvement through hybrid retrieval.

A topic entity refers to the main subject of a question, e.g., *'USA'* in *"Who's the president of the USA?"* The `Selected` strategy picks the better retriever for each question based on ground-truth answers, serving as an approximate upper bound. While the two base retrievers perform comparably, `Selected` substantially outperforms both on SKBs, underscoring the potential of hybrid retrieval.

Nevertheless, existing Hybrid-RAG works still face two critical challenges. *(1) Lack of Global Information.* Most works first identify topic entities within the query and subsequently apply ego-graph searching (Wang et al., 2023) or path searching (Ma et al., 2025) to leverage relational information. However, relevant entities and documents may be located at a considerable distance from the identified topic entities. The restriction on the locality of graph traversal limits the flexibility and effectiveness of retrieval. *(2) Lack of Query-Aware Retriever Selection.* Existing Hybrid-RAG methods typically retrieve both relational and textual information and concatenate them as input, regardless of their relevance to the current query (Sarmah et al., 2024). This indiscriminate fusion can confuse the LLM, as it may struggle to prioritize the retrieved documents between redundant or conflicting sources. Worse still, excessive inclusion of irrelevant text inflates the input length and increases reasoning complexity (Li et al., 2024b).

To overcome these limitations, we propose CoRAG, a novel Hybrid-RAG framework that adaptively combines textual search with graph-based relational search depending on the query. The core idea of CoRAG is to enable a multiple-retriever framework that autonomously integrates relational and textual information during each retrieval request. Unlike previous methods that are restricted to local graph neighborhoods, CoRAG can flexibly retrieve global relevant information through query-document text embedding matching. Moreover, CoRAG intelligently blends textual and relational retrieval results using a cooperative retriever architecture, ensuring a coherent and effective integration of diverse knowledge sources. Specifically, our hierarchical gating mechanism (Li et al., 2025) intelligently combines retrieval results from both textual and relational sources, thereby enhancing the model's overall performance and robustness. Overall, our contributions are as follows:

(1) We identify critical shortcomings in existing Hybrid-RAG methods, particularly their limited retrieval scope and ineffective integration of relational and textual information.

(2) We propose CoRAG, an intuitive yet powerful approach that dynamically selects retrieval sources and integrates textual and relational data seamlessly.

(3) Extensive experiments on diverse semi-structured QA benchmarks (e.g., medicine, e-commerce, and academic domains) demonstrate that CoRAG consistently outperforms baselines, highlighting its effectiveness and general applicability.

2 Preliminary

In this paper, we focus on the semi-structured knowledge base question answering (SKBQA) task (Wu et al., 2024b), which extends the traditional knowledge base question answering (KBQA) task (Talmor and Berant, 2018). In SKBQA, the LLMs must not only retrieve relevant information from the graph structures of SKB, but also from an external semi-structured knowledge base (SKB) by introducing entity-level documents. A detailed definition is provided below:

Definition 2.1 (SKBQA). *SKBQA consists of two components: (i) a semi-structured knowledge base (SKB) and (ii) a question-answering (QA) dataset.* ***SKB*** *is a document-enhanced knowledge graph* $SKB = \{\mathcal{E}, \mathcal{T}, \mathcal{D}, T_E, T_R\}$*, where* $\mathcal{E}, \mathcal{T}$*, and* $\mathcal{D}$ *denote the entity set, the triplet set, and the document set, respectively. Each entity* $e \in \mathcal{E}$ *is aligned with one descriptive document* $D_e \in \mathcal{D}$*.* T_E *and* T_R *represent the collections of all entity*

types and relation types, respectively. Each triplet $t \in \mathcal{T}$ can be represented as (e_1, r, e_2), where $\{e_1, e_2\} \subset \mathcal{E}, r \in T_R$.

***QA** dataset consists of question/answers pairs: $\mathcal{D}_{QA} = \{(q, \mathcal{A})\}$, where the question q appears in the form of a string (e.g., "What drugs target the CYP3A4 enzyme and are used to treat strongyloidiasis?"), and the answer set $\mathcal{A}$ is a set of entities (e.g., {"Ivermectin"} for the before-mentioned q) which satisfies $\mathcal{A} \subset \mathcal{E}$. Given a SKB and a $\mathcal{D}_{QA}$, the $SKBQA$ task requires the model to predict a set $\hat{\mathcal{A}} \subset \mathcal{E}$ consisting of k potential answer entities, aiming to maximize the coverage of all ground-truth answer entities: $\max_{\hat{\mathcal{A}}}(|\hat{\mathcal{A}} \cap \mathcal{A}|)$. We quantify the task objective using Hits@k metrics.*

3 Methodology

In this section, we introduce CoRAG, a Hybrid-RAG framework that dynamically integrates both relational knowledge and textual information. We begin by extracting topic entities (§ 3.1) from the question, which will be used in the subsequent retrieval process. Next, we design two distinct types of retrievers to separately fetch textual and relational information (§ 3.2). In Section 3.3, we present the core component of our approach, the Cooperative-Retrievers (CoR). As illustrated in Figure 2, CoR employs a hierarchical architecture to combine these two sources of information, which utilizes a gate network that dynamically assigns higher attention scores to the retrievers most relevant to answering the question. Lastly, we introduce CoRAG (§ 3.4), which integrates CoR with an LLM reranker, leveraging LLMs' reasoning strengths in the SKBQA task.

3.1 Topic Entity Extraction

The first step aims to find the most relevant entities mentioned in a user's query, which can be conceptualized as a named entity recognition task (NER) (Keraghel et al., 2024). In the context of SKBQA, the set of all entity names ($\mathcal{E}.names$) acts as the entity gazetteer, which has a limited size. Therefore, rule-based NER (Jehangir et al., 2023) is sufficient for the recognition task. As demonstrated in Algorithm 1, we employ the Aho-Corasick Automaton algorithm (Aho and Corasick, 1975) to identify named entities, utilizing greedy matching to prioritize entities with longer string names. Additionally, topic entity pruning via LLM will be incorporated as an optional step to eliminate erroneous named entity recognition results from the first-stage extraction. We denote the topic entities of question q as $\mathcal{E}^T$. In general QA tasks, it is typically assumed that each entity in the corresponding answer set $\mathcal{A}$ does not appear explicitly in the text of the question q. Therefore, we have the set of candidate entities: $\mathcal{C}_q = \mathcal{E} \backslash \mathcal{E}^T$, and $\mathcal{A} \subset \mathcal{C}_q$.

3.2 Information Retrievers

Given a question q, we design two types of base retrievers to handle textual and relational information separately. Both types of retrievers are implemented using a bi-encoder backbone R^*. Given a document $e \in \mathcal{C}_q$, we score it based on its semantic similarity with the question q:

$$s(q, e) = R^*(q, D_e) = \langle E_q(q), E_d(D_e)) \rangle, \quad (1)$$

where $\langle \cdot, \cdot \rangle$ represents the cosine-similarity function. E_q and E_d are two encoders for question and document embedding generation. The score $s(q, e)$ measures the relevance of the document to the question, which is further utilized for entity ranking in the final stage of retrieval. The key distinction between the two types of retrievers lies in the scope of entity documents they retrieve, which will be detailed in the following.

Textual Retriever. The textual retriever module $R^{(t)}$ is designed to retrieve from different types of entity documents. First, we categorize all nodes in the SKB into $n = |T_E|$ distinct groups based on their entity types. For each entity type $E \in T_E$, we assign an individual textual retriever $R^{(t)}_E$. Specifically, the retrieval scope is restricted to all entities of type E, while the relevance score is computed as follows:

$$R^{(t)}_E = \begin{cases} \langle E_q(q), E_d(D_e) \rangle, & T_e = E, \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where T_e represent the entity type of e. Since $R^{(t)}$ performs retrieval over the entire graph based on entity-level attributes (i.e., document content and entity types), the challenge of lacking global information is inherently mitigated.

Relational Retriever. The relational retriever $R^{(r)}$ is designed to retrieve local graph neighborhood entity documents related to the query q based on the structural connections within the SKB graph. Compared to textual retrievers, relational retrievers require the prior specification of the topic entities $\mathcal{E}^T$. During the retrieval process, each relational retriever $R^{(r)}_R$ exclusively scores the neighboring

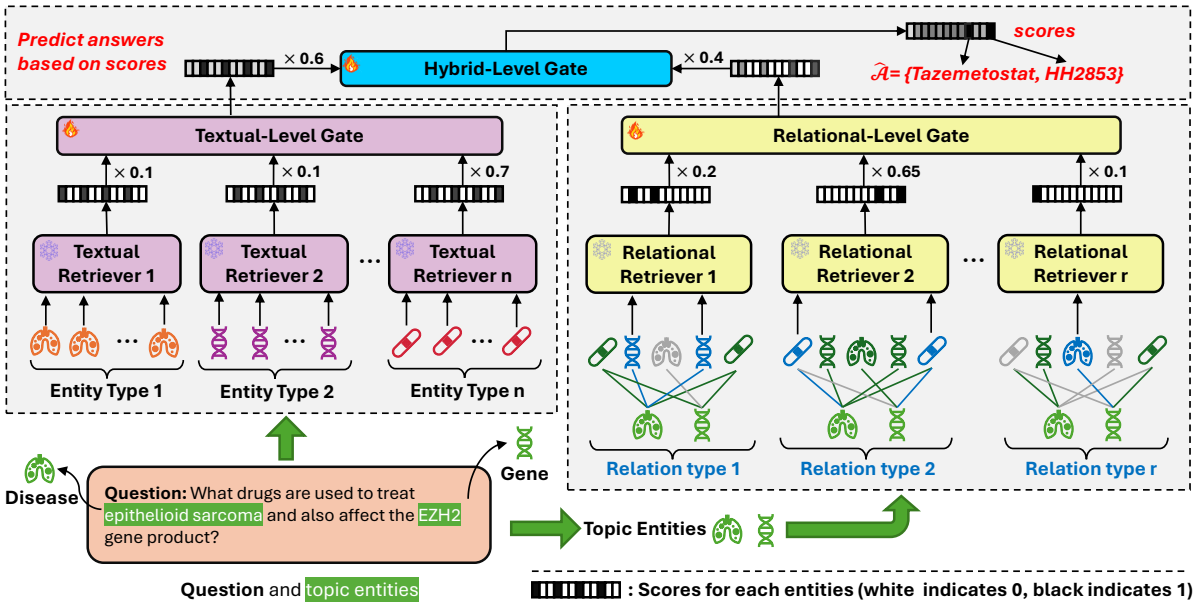

Figure 2: The architecture of Cooperative-Retrievers (CoR). n, r represent the number of entity types $|T_E|$ and relation types $|T_R|$, respectively. Each retriever's scope is restricted to a specific subset of the entity set $\mathcal{E}$, while the gate networks dynamically assign appropriate attentions to the retrievers based on the given question.

entity set $\mathcal{N}(R, \mathcal{E}^T)$ of $\mathcal{E}^T$ based on the relation type R:

$$\mathcal{N}(R, \mathcal{E}^T) = \{e | \exists s \in \mathcal{E}^T, (s, R, e) \in \mathcal{T}\} \backslash \mathcal{E}^T. \tag{3}$$

Consequently, the retrieval scope of the relational retriever $R_R^{(r)}$ is restricted to $\mathcal{N}(R, \mathcal{E}^T)$:

$$R_R^{(r)} = \begin{cases} \langle E_q(q), E_d(D_e) \rangle, & e \in \mathcal{N}(R, \mathcal{E}^T), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

3.3 Cooperative-Retrievers (CoR)

In this section, we build up a mixture-of-experts architecture to effectively integrate textual and relational retrieval results (challenge 2). Its core idea is simple yet powerful: each question may benefit differently from textual or relational information, and may focus on different types of entities or relations. Thus, CoR adaptively emphasizes the retriever type most relevant to a given query, effectively addressing the challenge of query-aware retriever selection.

3.3.1 CoR Structure

As shown in Figure 2, we adopt a hierarchical architecture to enhance the scalability of retriever selection for different $|T_E|$ and $|T_R|$. At the bottom of the entire structure, we assign a source-level gate to each source (textual/relational) that aggregates results from the corresponding type of retrievers. The output of all the source-level gates will be integrated by the top hybrid gate. Each gate network applies the softmax function for activation prior to generating its output.

Textual Gate. For a given question q, a candidate entity $e \in \mathcal{C}_q$, and $|T_E|$ textual retrievers for different candidate types, we first calculate the score vector $\mathbf{s}^{(t)}(q, e)$ for e which consists of scores generated by all the textual retrievers $R_E^{(t)}$:

$$\mathbf{s}^{(t)}(q, e) = [R_E^{(t)}(q, D_e)]_{E \in T_E}, \tag{5}$$

where T_E represents the set of all entity types. The textual gate $G^{(t)}$ is designed to assign appropriate weights to different textual retrievers, enabling adaptive integration of their scores based on the question q. Specifically, $G^{(t)}$ produces a weight vector, which is used to compute the final textual score $s^{(t)}$ through an inner product calculation:

$$s^{(t)}(q, e) = (G^{(t)}(q))^\top \mathbf{s}^{(t)}(q, e). \tag{6}$$

Relational Gate. As a source-level gate, the relational gate $G^{(r)}$ computes the relational score $s^{(r)}$ similarly to the textual gate $G^{(t)}$:

$$\mathbf{s}^{(r)}(q, e) = [R_R^{(r)}(q, D_e, \mathcal{E}^T)]_{R \in T_R}, \tag{7}$$

$$s^{(r)}(q, e) = (G^{(r)}(q))^\top \mathbf{s}^{(r)}(q, e), \tag{8}$$

where $\mathcal{E}^T$ represents the topic entities of q, $R_R^{(r)}$ is the relational retriever for relation type R.

Hybrid Gate. After the calculation of textual/relational scores $s^{(t)}$ and $s^{(r)}$, we utilize a hybrid gate $G^{(h)}$ to aggregate them and compute the predicted score $\hat{s}$. The architecture and inference of $G^{(h)}$ is identical to $G^{(t)}$ and $G^{(r)}$:

$$\mathbf{s}^{(h)}(q,e) = [s^{(t)}(q,e), s^{(r)}(q,e)], \quad (9)$$

$$\hat{s}(q,e) = (G^{(h)}(q))^{\top}\mathbf{s}^{(h)}(q,e), \quad (10)$$

where t, r represent the texutal and relational source. Eventually, the top-k candidate entities $\hat{\mathcal{A}}$ ranked by $\hat{s}(q,e)$ can either be directly used for evaluation or be reranked in § 3.4.

Training Objective. For a given QA pair $(q, \mathcal{A}) \in \mathcal{D}_{QA}$, we assign the ground-truth score $s(q,e)$ of a candidate entity e as 1 if $e \in \mathcal{A}$ otherwise 0. To minimize the mismatch between s and $\hat{s}$, the objective function is formulated as Mean Squared Error loss:

$$\mathcal{L}(q,\hat{s}) = \frac{1}{|\mathcal{C}_q|}\sum_{e\in\mathcal{C}_q}\left(\hat{s}(q,e) - s(q,e)\right)^2, \quad (11)$$

where $\mathcal{C}_q$ represents the set of candidate entities.

3.4 CoRAG

The preceding discussion elaborates on our retriever module CoR. We further integrate an LLM-based reranker lr within the Hybrid-RAG framework, thereby forming our complete CoRAG method. For a given question q, we first retrieve the top-k candidate entities $\hat{\mathcal{A}} = CoR(q, \mathcal{E}^T)$, where $\mathcal{E}^T$ represents the topic entities of q. We then ask the reranker lr to generate binary relevance scores for each $e \in \hat{\mathcal{A}}$:

$$s'(q,e) = lr(q, D_e) \in \{0,1\}. \quad (12)$$

We then fuse the scores from the reranker and CoR:

$$\hat{s}^*(q,e) = \hat{s}(q,e) + s'(q,e), \quad (13)$$

where $\hat{s}(q,e)$ is the score given by our CoR module. Given the bounded range of cosine similarity, the CoR score $\hat{s}(q,e)$ is guaranteed to lie within $[0,1]$. Such property ensures that entities with the reranker score $s' = 1$ are always ranked higher than those entities with $s' = 0$, effectively granting the reranker score s' precedence over the CoR score $\hat{s}$. Finally, CoRAG ranks the candidate entities $\mathcal{C}_q$ according to $\hat{s}^*(q,e)$. Please refer to Appendix A.2 for detailed prompt template of the reranker.

4 Experiments

In this section, we conduct experiments to evaluate the effectiveness of our proposed CoRAG. Our primary focus is to address the following research questions: **RQ1**: How is the performance of our CoRAG framework (and its retriever module CoR) on semi-structured knowledge base question-answer tasks, compared to other baselines? **RQ2**: How do the individual modules within our framework separately contribute to the overall performance? **RQ3**: How do the model's hyperparameters impact the performance of CoR? **RQ4**: Does CoRAG provide an interpretable selection of different textual and relational retrievers?

4.1 Empirical Settings

Implementation Details. In the CoRAG framework, we implement dense gating mechanisms using a two-layer MLP and employ softmax as the output activation function to dynamically allocate attention between retrievers. The input of our gate networks (discussed in § 3.3) is the question embeddings generated by `text-ada-002` from OpenAI.[2] Recall that we use the bi-encoder as the backbone architecture for all base retrievers in our framework. Unless stated otherwise, both the query (question) encoder E_q and the document encoder E_d are implemented using `text-ada-002`. Note that the encoder backbone are pretrained and fixed through out the entire framework.

Experimental Details. We implement and execute all code on a server equipped with 48-core CPUs and NVIDIA A100 GPUs. The framework is built using Python 3.11 and PyTorch 2.6.0. Unless otherwise specified, we set the number of training epochs to 5 and the learning rate to 1×10^{-4}. For our LLM-Reranker, we use `gpt-4o-mini` from OpenAI as the backbone model.

Datasets. We conduct experiments on three challenging real-world SKBQA datasets from STaRK (Wu et al., 2024b), comprising domains of e-Commerce (STaRK-Amazon), academics (STaRK-MAG), and medicine (STaRK-Prime). We provide basic statistics in Table 1. Detailed dataset examples are shown in Table 4.

Metrics. The empirical results are evaluated using Hits @ 1, Hits @ 5, Recall @ 20, and MRR (mean reciprocal rank) metrics.

Retriever Baselines. To enable a thorough and fair comparison for both our retriever module CoR and

[2] https://openai.com/index/new-and-improved-embedding-model/

Datasets	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert T_E\rvert$	$\lvert T_R\rvert$	$\overline{tok(D_e)}$	$\lvert\mathcal{D}_{\mathbf{QA}}\rvert$	$\overline{tok(q)}$	$\overline{\lvert\mathcal{A}\rvert}$
Amazon	1,035,542	9,443,802	4	5	546.23	9100	32.27	18.28
MAG	1,872,968	39,802,116	4	4	110.40	13323	32.48	2.78
Prime	129,375	8,100,498	10	18	204.48	11204	31.65	2.56

Table 1: The blocks in the middle and on the right, respectively, represent the statistics of the SKB and QA datasets. The 'STaRK-' prefix is omitted in the dataset names. The operation $|\mathcal{S}|$ denotes the size of the set $\mathcal{S}$, while $\overline{|\mathcal{A}|}$ represents the average answer entity numbers by each QA pair in the test QA dataset. $tok(text)$ denotes the token numbers of the text.

	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
Retriever Methods												
DPR (RoBERTa)	15.29	47.93	44.49	30.20	10.51	35.23	42.11	21.34	4.46	21.85	30.13	12.38
DPR (text-ada-002)	39.16	62.73	53.29	50.35	29.08	49.61	48.36	38.62	12.63	31.49	36.00	21.41
Multi-VSS	40.07	**64.98**	55.12	**51.55**	25.92	50.43	50.80	36.94	15.10	33.56	38.05	23.49
CoR	**40.99**	64.52	**55.75**	51.46	**43.00**	**60.15**	**56.98**	**50.52**	**20.85**	**40.49**	**47.44**	**30.02**
Relative Improvement	2.3%	-0.7%	1.1%	-0.2%	47.9%	19.3%	12.2%	30.8%	38.1%	20.7%	24.7%	27.8%
Retrieval-Augmented Generation (RAG) Methods												
ReAct	42.14	64.56	50.81	52.30	31.07	49.49	47.03	39.25	15.28	31.95	33.63	22.76
Reflexion	42.79	65.05	54.70	52.91	40.71	54.44	49.55	47.06	14.28	34.99	38.52	24.82
AVATAR-C	40.92	63.63	53.68	51.73	33.25	52.17	47.88	41.34	8.82	23.82	30.32	16.20
AVATAR	49.87	69.16	**60.57**	58.70	44.36	59.66	50.63	51.15	18.44	36.73	39.31	26.73
QAGNN	26.56	50.01	52.05	37.75	12.88	39.01	46.97	29.12	8.85	21.35	29.63	14.73
ToG	-	-	-	-	13.16	16.17	11.30	14.18	6.07	15.71	13.07	10.17
AGR	49.82	62.97	53.38	56.77	39.29	53.66	51.89	46.20	25.85	**44.41**	**46.63**	35.04
CoRAG	**49.88**	**70.04**	58.17	**59.31**	**49.42**	**64.20**	**58.66**	**55.95**	**28.45**	43.36	45.89	**35.15**
Relative Improvement	0.0%	1.3%	-4.0%	1.0%	11.4%	7.6%	13.1%	9.4%	10.1%	-2.4%	-1.6%	0.3%

Table 2: Main results across STaRK-Amazon, STaRK-MAG, and STaRK-Prime datasets. **CoR** represents the Cooperative-Retrievers module in our framework **CoRAG**. The best and runner-up models are respectively highlighted in **bold** and underlined. We also report the relative improvement of our methods compared to the best retriever or RAG baseline, respectively.

the complete RAG framework CoRAG, we include two categories of baselines. In the retrieval setting, where the answers are directly retrieved from the knowledge base, we incorporate embedding-based retrieval approaches (DPR (Karpukhin et al., 2020), and Multi-VSS (Wu et al., 2024a)) as baselines for the comparison with CoR.

RAG Baselines. In the RAG setting, where the answers are generated by LLMs, we incorporate both traditional RAG methods (ReAct (Yao et al., 2022), Reflextion (Shinn et al., 2023), AGR (Chen et al., 2024), and AvaTaR (Wu et al., 2024a)) and Graph RAG methods (QAGNN (Yasunaga et al., 2021), ToG (Sun et al., 2024)). Notably, both ReAct and AvaTaR utilize the STaRK library's API for graph traversal during the reasoning process, ensuring a fair comparison with CoRAG. Detailed baseline settings are shown in Appendix A.1.

4.2 Main Results (RQ1)

To answer RQ1, we compare retriever baselines with CoR and RAG baselines with CoRAG. For each question q, the top-25 entities retrieved by CoR are processed by the LLM-Reranker. The empirical results from the SKBQA task, presented in Table 2, highlight the following key observations: (1) Our Cooperative-Retrievers module significantly outperforms both DPR and Multi-VSS, underscoring the effectiveness and necessity of integrating relational information into the retrieval process. (2) With a straightforward LLM-Reranker, CoRAG surpasses existing RAG methods across almost all metrics, demonstrating that the primary bottleneck in current Hybrid-RAG applications lies

in the retriever module's performance. (3) CoR's relative improvements on STaRK-MAG (academic) and STaRK-Prime (medical) are more pronounced, suggesting that tasks requiring deeper semantic reasoning and knowledge synthesis benefit significantly from the incorporation of relational information. (4) In contrast, CoRAG shows less substantial performance gains over the baselines on STaRK-Amazon. This may be attributed to the fact that recommendation tasks in the E-Commerce domain are more reliant on product attributes and textual descriptions than on the logical relationships present in the knowledge base.

4.3 Ablation Study (RQ2)

To answer RQ2, we systematically remove each of the three sub-modules in our method to evaluate their individual contributions: (1) w/o Textual Retrievers. In this ablation, we remove textual retrievers from our framework and only use relational retrievers and the output score $s^{(r)}$ from the relational gate for inference. (2) w/o Relational Retrievers. Similarly, we only keep textual retrievers and the textual gate in this ablation. (3) w/o Hybrid Gate. For this analysis, instead of utilizing the hybrid gate network $G^{(h)}$ in Eq. 10, we directly compute the average of the textual score $s^{(t)}$ (Eq. 6) and relational score $s^{(r)}$ (Eq. 8) for evaluation.

The experimental results in Table 3 provide the following insights: (1) The effectiveness of our model design is validated by the performance decline observed across all datasets when any submodule is removed. (2) Without the retrieval information from the textual retrievers, the evaluation metrics exhibit a substantial decline across all scenarios, emphasizing that it cannot be substituted by relational retrievers. (3) Without relational retrievers, the model also experiences a performance decline, with a particularly severe drop on Prime. This underscores the varying importance of relational information across different domains. (4) The hybrid gate provides a slight performance boost on the Amazon dataset, confirming our observation in the main results that relational information offers limited improvement in this dataset.

4.4 Sensitivity Analysis (RQ3)

To address RQ3, we examine the impact of model scale on performance. Additionally, we analyze the API usage of the LLM-Reranker within CoR.

Model Scale. In this subsection, our objective is to evaluate the impact of model scale on CoR.

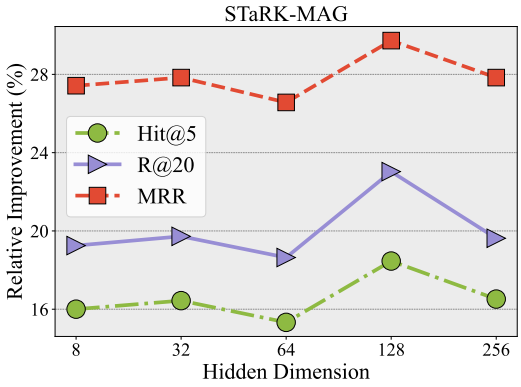

Figure 3: An intuitive analysis of hybrid retrieval on the STaRK-MAG and STaRK-Prime datasets.

Specifically, we adjust the model size of the textual, relational, and hybrid gate networks, varying the hidden dimensions (while maintaining consistency across all gate networks). We perform experiments on the STaRK-MAG dataset and present the relative improvement of CoR over the best results of embedding-based retrieval approaches in Figure 3. From the results, we observe the following insights: (1) CoR performance consistently outperforms baselines in different model sizes ($> 16\%$), indicating its robustness and scalability to varying model scales. (2) As the size of the model increases, the performance initially improves as the model is able to capture more information. However, beyond a certain point, performance begins to decline due to insufficient training.

4.5 Case Study (RQ4)

To answer RQ4, we illustrate examples of CoR inference in Figure 4. The query is asking for a '*cellular component*'/'*pathway*' with certain properties, and it also describes a *biological process* through phrases like 'involving... results in... through...'. We observe that the textual gate assigns high scores to these two entity types, encouraging the inclusion of textual retrievers for these types. Regarding relational retrieval, if congenital adrenal insufficiency is a phenotype of FDX2 dysfunction, the triplet (FDX2, phenotype present, adrenal insufficiency) needs to be given more attention. Additionally, understanding the impact of lost FDX2 expression can also help answer the question. As a result, the relational gate assigns high scores to retrievers like 'phenotype_present' and 'expression_absent'. This case study demonstrates the interpretability of our CoR design, as the trained gate models can assign high attention scores to the retrievers that

	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
CoR w/o TR	21.62	33.62	31.33	27.15	31.93	40.64	38.96	35.69	15.49	26.42	31.74	20.35
CoR w/o RR	39.71	63.40	56.11	50.67	29.08	49.61	50.12	38.30	8.64	19.53	22.97	13.60
CoR w/o HG	40.68	64.07	**58.21**	51.32	42.29	59.32	**63.32**	49.92	14.92	25.38	30.75	19.72
CoR	**40.99**	**64.52**	55.75	**51.46**	**43.00**	**60.15**	56.98	**50.52**	**20.85**	**40.49**	**47.44**	**30.02**

Table 3: Ablation study results. Here, TR, RR, and HG denote the Textual Retriever, Relational Retriever, and Hybrid Gate, respectively. The best and runner-up models are respectively highlighted in **bold** and underlined.

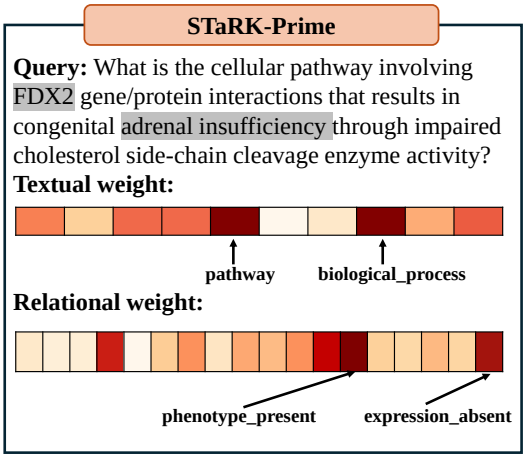

Figure 4: Case study of gate models in CoR. We present a heatmap of attention scores in the textual and relational gates. The labels on the heatmap correspond to the label order in Table 4. Shaded text represents topic entities.

contribute most to answering the question.

5 Related Works

5.1 Retrieval Augmented generations

Retrieval-Augmented Generation (RAG) is a technique that integrates retrieval mechanisms with generative models to improve the quality of generated responses (Fan et al., 2024). The retriever model retrieves relevant information from a knowledge base, which is then utilized by a generative model (e.g., large language models (Touvron et al., 2023)) to produce more contextually accurate and informative output. Within RAG, Graph-RAG (Peng et al., 2024) (and Hybrid-RAG (Yuan et al., 2024; Sarmah et al., 2024)) incorporates graph-based structures to represent relationships between documents. By leveraging graphs, RAG models can exploit richer contextual connections between retrieved documents, enhancing the coherence and informativeness of generated responses. Some models (Wang et al., 2023) include a graph construction phase, where documents are linked based on semantic or structural relationships.

5.2 Ensemble Retrievers

Ensemble retrieval methods (Li et al., 2024a) improve performance by combining outputs from multiple retrieval models, typically via *score fusion*, *rank fusion*, or *multi-stage retrieval*. Score fusion (Fox and Shaw, 1994) aggregates scores from different retrievers into a unified score per document, using methods ranging from simple averaging or weighted sums (Guo et al., 2025) to learned fusion functions (Khattab and Zaharia, 2020). Rank fusion (Cormack et al., 2009) integrates ranked lists by focusing on document order rather than scores. Techniques like Borda Count (Fox and Bruyns, 2025) assign points based on rank positions to compute a final ranking. Multi-stage retrieval applies sequential retrieval steps, typically culminating in a reranking phase (Liu et al., 2022; Gao et al., 2025), where documents are reordered using additional models or signals. Our framework adopts a multi-stage architecture with an LLM-based reranker, further extending ensemble strategies to incorporate graph-structured knowledge.

6 Conclusion

We propose CoRAG, a Hybrid Retrieval-Augmented Generation framework that dynamically integrates relational and textual knowledge through a Cooperative-Retrievers architecture. By enabling global retrieval beyond local graph neighborhoods and adaptively fusing heterogeneous information, CoRAG overcomes key limitations of existing Hybrid-RAG methods. Experiments on semi-structured knowledge base QA benchmarks demonstrate that CoRAG achieves state-of-the-art performance, outperforming both standard RAG and graph-based RAG methods. Our results emphasize the importance of adaptive retrieval selection in enhancing Hybrid-RAG performance.

7 Limitations & Future Works

One key constraint of our CoR module is its ineffectiveness in handling complex logical queries that require multi-hop reasoning. This limitation primarily arises from the fact that our training process relies solely on supervision from ground-truth answers. Feasible solutions include: (1) Incorporate additional supervision by leveraging LLMs to generate annotated reasoning steps. (2) Utilizing reinforcement learning and reward shaping techniques to generate intermediate supervision signals for multi-step reasoning. Additionally, more advanced models can be employed for topic entity extraction. For future works, we aim to refine retriever selection strategies, explore multi-hop retrieval paths, and extend the framework to domain-specific applications requiring complex reasoning.

8 Ethics Statement

This paper proposes CoRAG, a hybrid retrieval augmented generation framework that can effectively retrieve accurate answers in SKBQA tasks. While acknowledging the need for responsible usage of the proposed method, we do not foresee major negative societal impacts.

Acknowledgement

This work is supported in part by the National Science Foundation (NSF) under grants IIS-2006844, IIS-2144209, IIS-2223769, CNS-2154962, BCS-2228534, and CMMI-2411248; the Office of Naval Research (ONR) under grant N000142412636; and the Commonwealth Cyber Initiative (CCI) under grant VV-1Q24-011.

References

Alfred V. Aho and Margaret J. Corasick. 1975. Efficient string matching: an aid to bibliographic search. *Commun. ACM*, 18(6):333–340.

Xinran Chen, Xuanang Chen, Ben He, Tengfei Wen, and Le Sun. 2024. Analyze, generate and refine: Query expansion with LLMs for zero-shot open-domain QA. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11908–11922, Bangkok, Thailand. Association for Computational Linguistics.

Zihan Chen, Song Wang, Zhen Tan, Xingbo Fu, Zhenyu Lei, Peng Wang, Huan Liu, Cong Shen, and Jundong Li. 2025. A survey of scaling in large language model reasoning. *Preprint*, arXiv:2504.02181.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Xavier Daull, Patrice Bellot, Emmanuel Bruno, Vincent Martin, and Elisabeth Murisasco. 2023. Complex qa and language models hybrid architectures, survey. *Preprint*, arXiv:2302.09051.

Dennis Diefenbach, Vanessa López, Kamal Deep Singh, and Pierre Maret. 2018. Core techniques of question answering systems over knowledge bases: a survey. *Knowl. Inf. Syst.*, 55(3):529–569.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. *Preprint*, arXiv:2405.06211.

Edward Fox and Joseph Shaw. 1994. Combination of multiple searches. *NIST special publication SP*, pages 243–243.

N. Bradley Fox and Benjamin Bruyns. 2025. An evaluation of borda count variations using ranked choice voting data. *Preprint*, arXiv:2501.00618.

Jingtong Gao, Bo Chen, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2025. Llm4rerank: Llm-based auto-reranking framework for recommendations. *Preprint*, arXiv:2406.12433.

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

Jiafeng Guo, Yinqiong Cai, Keping Bi, Yixing Fan, Wei Chen, Ruqing Zhang, and Xueqi Cheng. 2025. Came: Competitively learning a mixture-of-experts model for first-stage retrieval. *ACM Trans. Inf. Syst.*, 43(2).

Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. 2023. A survey on named entity recognition — datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent advances in named entity recognition: A comprehensive survey and comparative study. *Preprint*, arXiv:2401.10825.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Preprint*, arXiv:2004.12832.

Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N. Ioannidis, Huzefa Rangwala, and Christos Faloutsos. 2024. Hybgrag: Hybrid retrieval-augmented generation on textual and relational knowledge bases. *Preprint*, arXiv:2412.16311.

Mingda Li, Xinyu Li, Yifan Chen, Wenfeng Xuan, and Weinan Zhang. 2024a. Unraveling and mitigating retriever inconsistencies in retrieval-augmented large language models. *Preprint*, arXiv:2405.20680.

Weikai Li, Ding Wang, Zijian Ding, Atefeh Sohrabizadeh, Zongyue Qin, Jason Cong, and Yizhou Sun. 2025. Hierarchical mixture of experts: Generalizable learning for high-level synthesis. *Preprint*, arXiv:2410.19225.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024b. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. *Preprint*, arXiv:2305.13269.

Yihao Li, Ru Zhang, and Jianyi Liu. 2024c. An enhanced prompt-based llm reasoning scheme via knowledge graph-integrated collaboration. *Preprint*, arXiv:2402.04978.

Zhuoyang Li, Liran Deng, Hui Liu, Qiaoqiao Liu, and Junzhao Du. 2024d. Unioqa: A unified framework for knowledge graph question answering with large language models. *Preprint*, arXiv:2406.02110.

Weiwen Liu, Yunjia Xi, Jiarui Qin, Fei Sun, Bo Chen, Weinan Zhang, Rui Zhang, and Ruiming Tang. 2022. Neural re-ranking in multi-stage recommender systems: A review. *Preprint*, arXiv:2202.06602.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Pei-Chi Lo and Ee-Peng Lim. 2023. Contextual path retrieval: A contextual entity relation embedding-based approach. *ACM Trans. Inf. Syst.*, 41(1).

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. *Preprint*, arXiv:2407.10805.

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

Bhaskarjit Sarmah, Benika Hall, Rohan Rao, Sunil Patel, Stefano Pasquali, and Dhagash Mehta. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. *Preprint*, arXiv:2408.04948.

Lukas Schmelzeisen, Corina Dima, and Steffen Staab. 2021. Wikidated 1.0: An evolving knowledge graph dataset of wikidata's revision history. *Preprint*, arXiv:2112.05003.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

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

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep

self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2023. Knowledge graph prompting for multi-document question answering. *Preprint*, arXiv:2308.11730.

Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis N. Ioannidis, Karthik Subbian, Jure Leskove, and James Zou. 2024a. Avatar: Optimizing llm agents for tool usage via contrastive reasoning.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N. Ioannidis, Karthik Subbian, James Zou, and Jure Leskovec. 2024b. Stark: Benchmarking llm retrieval on textual and relational knowledge bases. In *NeurIPS Datasets and Benchmarks Track*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

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

Yizhen Zheng, Huan Yee Koh, Maddie Yang, Li Li, Lauren T. May, Geoffrey I. Webb, Shirui Pan, and George Church. 2024. Large language models in drug discovery and development: From disease mechanisms to clinical trials. *Preprint*, arXiv:2409.04481.

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025. Collaborative retrieval for large language model-based conversational recommender systems. In *WWW*, pages 3323–3334.

A Implementation Details

A.1 Baselines

DPR and Multi-VSS. We implement both baselines with the `text-ada-002` model from OpenAI as the language encoder backbones. For DPR, we also include a variation using *roberta-base* (Liu et al., 2019) from the HuggingFace library as the backbone.

AvaTaR/AvaTaR-C/ReAct/Reflextion. We directly follow the implementation in the AvaTaR paper (Wu et al., 2024a), in which AVATAR-C removes the comparator from the optimization pipeline.

A.2 Prompts

LLM Reranker. "You are a helpful assistant tasked with determining whether an <entity_type> satisfies a given query. Assign a score of 1 if there is clear and explicit evidence that the <entity_type> satisfies the query. Otherwise, assign a score of 0. Query: <query>. Information about the <entity_type>: <entity_doc>. Output: The numeric score for this entity is:"

B Supplementary Experiments

B.1 Encoder Backbones

To examine the generality of our framework on different text encoder models beyond `text-ada-002`, we separately employ three additional embedding models (MiniLM-L6, -L12 (Wang et al., 2020), and mpnet-base (Song et al., 2020)) as the backbone to evaluate dense retrieval (DPR) and our cooperative retrievers (CoR) on each dataset.

The comparison results are demonstrated in Table 5, which shows that our CoR method consistently outperforms dense retrieval, demonstrating its effectiveness and generalizability.

B.2 Human-Generated Queries

To examine the generality of our framework on different data distributions, we evaluate our CoR model on the split of human-generated queries in the STaRK dataset. Through the empirical results in Table 6, our framework consistently outperforms dense retrieval.

Algorithm 1 Topic Entities Exaction

Require: Entity set $\mathcal{E}$, QA dataset $\mathcal{D}_{QA}$, Aho-Corasick Automaton A
Ensure: Topic Entities $\mathcal{E}^T$ for each $q \in \mathcal{Q}$

A.build($\mathcal{E}$.names) ▷ Initialize Automaton A
$\mathcal{E}^T \leftarrow empty\_dictionary$
for each $(q, \mathcal{A}) \in \mathcal{D}_{QA}$ **do**
    $S \leftarrow \mathcal{A}$.search(q) ▷ $S = \{(s_i, e_i, e)\}$
    Sort S by s_i ascending, $|e|$ descending
    $\mathcal{E}^T(q) \leftarrow \emptyset$, $last\_end \leftarrow -1$
    for each $(s_i, e_i, e) \in S$ **do**
        if $s_i > last\_end$ **then**
            $\mathcal{E}^T(q) \leftarrow \mathcal{E}^T(q) \cup \{e\}$
            $last\_end \leftarrow e_i$
        end if
    end for
    $\mathcal{E}^T(q) \leftarrow prune(\mathcal{E}^T(q))$ ▷ (Optional)
end for
return $\mathcal{E}^T$

Algorithm 2 Analysis of Hybrid Retrieval Performance Upper Bound

Require: $\mathcal{D}_{QA}$: The test QA dataset.
Require: SKB: The semi-structured knowledge base
Require: $candidates$: Candidate entities list sorted by $similarity(q, D_e)$ for each query.
Require: $\mathcal{E}^T$: Topic Entities for each question q
Ensure: $hits1, hits5, recall, mrr$: Evaluation metrics for each methods.

for each $(q, \mathcal{A})$ in $\mathcal{D}_{QA}$ **do**
    $Textual\_top\_k(q) \leftarrow \{top\ k\ candidates\ from\ candidates(q)\}$
    $seeds \leftarrow \mathcal{E}^T(q)$
    $neighbors \leftarrow \emptyset$
    for each s in $\mathcal{E}^T$ **do**
        $neighbors \leftarrow neighbors \cup get\_neighbors(SKB, s)$
    end for
    Sort $neighbors$ by index order from $candidates(q)$
    $Relational\_top\_k(q) \leftarrow \{top\ k\ elements\ of\ sorted\ neighbors\}$
    if $|Textual\_top\_k(q) \cap \mathcal{A}| > |Relational\_top\_k(q) \cap \mathcal{A}|$ **then**
        $Best\_top\_k(q) \leftarrow Textual\_top\_k(q)$
    else if $|Textual\_top\_k(q) \cap \mathcal{A}| == |Relational\_top\_k(q) \cap \mathcal{A}|$ **then**
        $Best\_top\_k(q) \leftarrow Relational\_top\_k(q)$
    else
        $Best\_top\_k(q) \leftarrow Relational\_top\_k(q)$
    end if
end for
Compute evaluation metrics for $Best\_top\_k$, $Relational\_top\_k$, and $Textual\_top\_k$.

Dataset	Attributes	Attribute Values
STaRK-Prime	Entity Types T_E	{disease, gene/protein, molecular_function, drug, pathway, anatomy, effect/phenotype, biological_process, cellular_component, exposure}
	Relation Types T_R	{ppi, carrier, enzyme, target, transporter, contraindication, indication, off-label use, synergistic interaction, associated with, parent-child, phenotype absent, phenotype present, side effect, interacts with, linked to, expression present, expression absent}
	Example Question q	Can you supply a compilation of genes and proteins associated with endothelin B receptor interaction, involved in G alpha (q) signaling, and contributing to hypertension and ovulation-related biological functions?
	Example Answers $\mathcal{A}$	{EDN1, EDN2}
STaRK-MAG	Entity Types T_E	{author, institution, field_of_study, paper}
	Relation Types T_R	{author___affiliated_with___institution, paper___cites___paper, paper___has_topic___field_of_study, author___writes___paper}
	Example Question q	Find publications from Carma researchers that report detections using the Australian Square Kilometre Array Pathfinder (ASKAP) radio telescope.
	Example Answers $\mathcal{A}$	{ASKAP HI imaging of the galaxy group IC 1459, Wide-field broad-band radio imaging with phased array feeds: a pilot multi-epoch continuum survey with ASKAP-BETA, The Detection of an Extremely Bright Fast Radio Burst in a Phased Array Feed Survey}
STaRK-Amazon	Entity Types T_E	{product, brand, category, color}
	Relation Types T_R	{also_buy, also_view, has_brand, has_category, has_color}
	Example Question q	Looking for a chess strategy guide from The House of Staunton that offers tactics against Old Indian and Modern defenses. Any recommendations?
	Example Answers $\mathcal{A}$	{Beating the King's Indian and Benoni Defense with 5. Bd3}

Table 4: Detailed description for the datasets (SKBs and QA datasets) we used in the experiments.

	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
MiniLM-L6 (DPR)	30.14	51.76	40.97	40.50	23.00	39.55	39.78	31.16	8.28	19.06	22.88	13.68
MiniLM-L6 (CoR)	38.35	55.57	57.24	46.23	25.40	43.73	38.79	33.59	13.57	27.70	36.32	20.31
mpnet-base (DPR)	30.20	54.02	44.52	41.41	25.55	44.35	44.45	34.43	7.67	19.63	24.18	13.63
mpnet-base (CoR)	40.79	57.37	60.33	48.45	27.47	48.23	42.57	36.75	13.64	26.88	35.67	19.91
MiniLM-L12 (DPR)	9.01	22.41	19.43	15.81	10.81	21.35	23.46	16.27	0.89	2.39	2.67	1.81
MiniLM-L12 (CoR)	31.03	45.89	47.38	37.61	15.10	28.68	27.71	21.67	6.50	14.89	21.36	10.54

Table 5: Performance comparison of DPR (Dense Passage Retrieval) and our method CoR using different encoders (MiniLM-L6, mpnet-base, MiniLM-L12) on Amazon, MAG, and Prime datasets.

	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
DPR	27.16	45.68	17.44	35.50	20.24	28.57	25.59	24.92	12.24	26.53	31.40	19.54
CoR	28.40	48.15	22.24	36.66	26.19	33.33	32.10	30.39	19.39	30.61	38.83	24.82

Table 6: Comparison between DPR (Dense Passage Retrieval) and CoR on the human-generated evaluation sets across three datasets.