Knowledge-Aware Query Expansion with Large Language Models for Textual and Relational Retrieval

Yu Xia[1] **Junda Wu**[1] **Sungchul Kim**[2] **Tong Yu**[2]
Ryan A. Rossi[2] **Haoliang Wang**[2] **Julian McAuley**[1]
[1]University of California San Diego [2]Adobe Research
{yux078, juw069, jmcauley}@ucsd.edu
{sukim, tyu, ryrossi, hawang}@adobe.com

Abstract

Large language models (LLMs) have been used to generate query expansions augmenting original queries for improving information search. Recent studies also explore providing LLMs with initial retrieval results to generate query expansions more grounded to document corpus. However, these methods mostly focus on enhancing textual similarities between search queries and target documents, overlooking document relations. For queries like "Find me a highly rated camera for wildlife photography compatible with my Nikon F-Mount lenses", existing methods may generate expansions that are semantically similar but structurally unrelated to user intents. To handle such semi-structured queries with both textual and relational requirements, in this paper we propose a knowledge-aware query expansion framework, augmenting LLMs with structured document relations from knowledge graph (KG). To further address the limitation of entity-based scoring in existing KG-based methods, we leverage document texts as rich KG node representations and use document-based relation filtering for our **K**nowledge-**A**ware **R**etrieval (KAR). Extensive experiments on three datasets of diverse domains show the advantages of our method compared against state-of-the-art baselines on textual and relational semi-structured retrieval.

1 Introduction

Large language models (LLMs) have been utilized to expand original queries with additional contexts, capturing similar semantics of target documents and hence improving retrieval performance (Gao et al., 2023; Wang et al., 2023). While direct generations of LLMs introduce problems such as hallucination, out-dated information, and lack of domain knowledge, recent methods (Jagerman et al., 2023; Lei et al., 2024; Shen et al., 2024) explore augmenting LLMs with initial retrievals as contexts, e.g., pseudo relevance feedback (PRF) (Lv and Zhai, 2010; Li et al., 2022), to generate expansions more grounded to domain-specific document corpus.

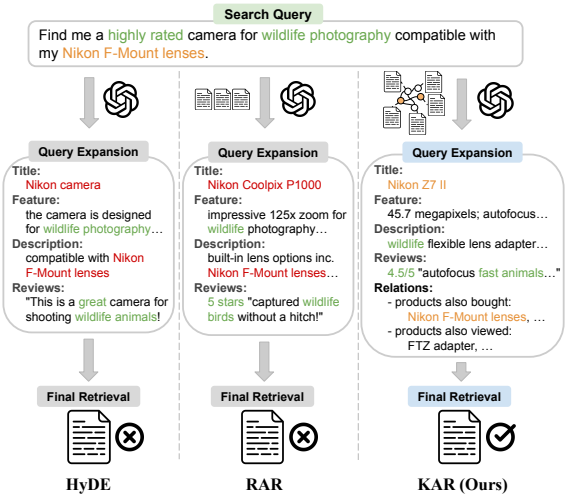

Figure 1: Example query expansions generated by HyDE (Gao et al., 2023), RAR (Shen et al., 2024), and KAR (Ours) given a semi-structured product search query with both textual and relational requirements (Wu et al., 2024b). While HyDE and RAR enrich the textual information, e.g., "wildlife" and "highly rated", they make up incorrect document relations, e.g., compatibility of "Nikon Coolpix P1000" with "F-Mount lenses". In contrast, our KAR utilizes document relations from knowledge graph, e.g., customers bought "Nikon Z7 II" and "F-Mount lenses" together, to generate semantically similarly and structurally related query expansions.

Though effective, existing methods mostly focus on enhancing the semantic similarities between expanded queries and target document texts. However, in real-world search scenarios, besides textual descriptions, documents are often inter-connected with certain types of relations (Talmor and Berant, 2018; Cao et al., 2022). Both textual and relational details are often queried by users in semi-structured manner (Wu et al., 2024b; Patel et al., 2024; Wu et al., 2024a; Boer et al., 2024) to help better describes their search intents, which are overlooked by existing query expansion methods. For exam-

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4275–4286

ple, the user query in Figure 1 specifies both textual requirements of the product "highly rated" and "wildlife photography", and relational requirement of the product "compatible with Nikon F-Mount lenses". While existing methods may generate semantically similar expansions on this product, they tend to make up incorrect product relations, i.e., compatibility between cameras and lenses, leading to suboptimal retrieval results.

To handle such semi-structured queries, we propose a knowledge-aware query expansion framework, augmenting LLMs with structured document relations from knowledge graphs (KG). We first parse entities explicitly mentioned the original query with an LLM and then retrieve textual documents of these entities as well as their associated nodes and relations on KG. While existing KG-based question answering methods (Yasunaga et al., 2021; Zhang et al., 2022; Taunk et al., 2023) filter out irrelevant relations by scoring the node relevance based on semantic similarity between the query and entity names, e.g., "Nikon camera", they overlook the rich textual details of entities queried by users, e.g., "highly rated" and "wildlife".

To address this, we leverage document texts as rich KG node representations and use document-based relation filtering to extract query-focused relations. Then, with collected textual and relational knowledge as inputs, LLM generates query expansions that are grounded to the document corpus while preserving user-specified document relations. The expanded queries are then utilized for the final retrieval as our **K**nowledge-**A**ware **R**etrieval (KAR). Extensive experiments are conducted on three textual and relational semi-structured retrieval datasets in the STaRK benchmark (Wu et al., 2024b) for product, academic paper, and biomedical search, respectively. The results show that our method outperforms state-of-the-art query expansion methods and achieves at least on par performances compared to LLM-based retrieval agent.

In summary, we make the following contributions: i) To handle complex search queries with both textual and relational requirements, we propose a knowledge-aware query expansion framework augmenting LLMs with KG; ii) To address the limitation of entity-based scoring, we use document texts as KG node representations and adopt document-based relation filtering for **K**nowledge-**A**ware **R**etrieval (KAR); iii) Experiments on three semi-structured retrieval datasets show the advantages of our method and its practical applicability.

2 Related Work

2.1 LLM-based Query Expansion

Query expansion has been a widely adopted technique in information search applications (Azad and Deepak, 2019), which expands the original query with additional contexts to match target documents. Earlier studies use initially retrieved documents as pseudo-relevance feedback (PRF) (Yu et al., 2003; Cao et al., 2008; Lv and Zhai, 2010; Li et al., 2022), extracting relevant content as supplemental information. However, the effectiveness of these methods are limited by the quality of initial retrievals.

Recently, LLM-enhanced information retrieval has been a prominent area (Zhu et al., 2023), where LLMs have been utilized to generate query expansions with their intrinsic knowledge. HyDE (Gao et al., 2023) employs an LLM to directly generate hypothetical documents that answer the query and then uses embeddings of them to retrieve similar real documents. Query2Doc (Wang et al., 2023) further improve the expansion quality by providing LLM with few-shot examples. Jagerman et al. (2023) also explore the use of chain-of-thought as expansions. To address the limitation that LLMs may lack domain-specific knowledge, Shen et al. (2024) propose retrieval-augmented retrieval (RAR) using initial retrievals as contexts for LLMs to generate query expansions. Lei et al. (2024) employ the LLM to first extract key information from initial retrievals before expanding the query. AGR (Chen et al., 2024) design Analyze-Generate-Refine, a multi-step query expansion framework, to incorporate LLMs' self-refinement ability with initial retrievals as references. Similar verification strategy is also explored in Jia et al. (2024). Despite these advances, existing methods mostly focus on textual similarities and overlook document relations. In comparison, our knowledge-aware query expansion augments LLMs with structured document relations from KGs for handling semi-structured retrieval tasks.

2.2 KG-Augmented LLM

In earlier studies, language models are used to provide text embeddings to enhance graph neural networks on KG reasoning tasks (Feng et al., 2020; Lin et al., 2021; Yasunaga et al., 2021; Spillo et al., 2023). With the emergent reasoning ability of LLMs over various textual structures, recently KG has in turn been utilized as a structured knowledge source to augment LLMs with factual or domain-

specific information for more grounded reasoning and generations (Pan et al., 2024). For example, Think-on-Graph (Sun et al., 2024) conducts entity and relation explorations to retrieve relevant triples for question answering. Reason-on-Graph (Luo et al., 2024) retrieves reasoning paths from KGs for LLMs to conduct faithful reasoning. HyKGE (Jiang et al., 2024) generates hypothesis reasoning paths to be grounded on KGs for answer generation. LPKG (Wang et al., 2024) constructs planning data from KGs for complex question answering. There is also a recent surge in graph retrieval augmented generation (Peng et al., 2024; Xu et al., 2024; He et al., 2024; Hu et al., 2024), which utilizes graph data such KGs as retrieval source for more accurate and structured response generation.

Compared to these studies on LLMs with KGs, our work differs in two key aspects. First, most prior studies focus on knowledge graph question answering, where queries are more fact-focused and answers are precisely encoded in KG. Our work focuses on document retrieval, where queries tend to be more descriptive and domain-specific. Second, in prior studies KG is the sole retrieval source. In our work, textual documents and knowledge graph serve as a semi-structured knowledge source with information covering diverse aspects (Wu et al., 2024b). Such semi-structured nature in retrieval source introduce distinct challenges for effective knowledge extraction and retrieval (Patel et al., 2024; Wu et al., 2024a; Boer et al., 2024).

3 Problem Definition

We define our studied query expansion for textual and relational semi-structured retrieval as follows. Follwing Wu et al. (2024b), suppose a knowledge base contains a collection of textual documents $\mathcal{D}$ and a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, where $d_i \in \mathcal{D}$ is a textual document describing an entity i, and $v_i \in \mathcal{V}$ is the corresponding entity node on KG, with $\mathcal{R}$ being a set of relations between different nodes. For example, in a paper search scenario as Figure 2, each paper i has a textual document d_i that contains abstract, venue, publication date, etc., and the corresponding node v_i on the knowledge graph $\mathcal{G}$ encodes its relations with other nodes such as paper citations, authorship, and field of study.

Now given a query q with requirements from both unstructured texts in $\mathcal{D}$ and structured relations in $\mathcal{G}$, the semi-structured retrieval (Wu et al., 2024b; Boer et al., 2024; Wu et al., 2024a) is to output a set of documents $\mathcal{A} \subseteq \mathcal{D}$ such that the entity described in each document satisfies both textual and relational requirements specified by query q. To bridge the gap between query and documents, we aims to augment the original query q with query expansions $\mathcal{Q}_e$ based on available textual and relational knowledge as

$$\mathcal{Q}_e = f(q, \mathcal{D}, \mathcal{G}), \tag{1}$$

$$q' = \text{Concat}(q, \mathcal{Q}_e), \tag{2}$$

where f represents a query expansion function and the expansions are then appended to q as the expanded query q' for the final document retrieval.

4 Methodology

In this section, we describe our knowledge-aware query expansion framework as in Figure 2.

Entity Parsing by LLM As the initial step, we first utilize an LLM to extract explicitly mentioned entities from the original query q given the document structures, denoted by $\mathcal{E}_q$. Following similar ideas in Gao et al. (2023), we also consider the original query q itself as a pseudo entity representing the target entity document to be retrieved,

$$\mathcal{E}_q = \{q,\ \text{LLM}(q)\}. \tag{3}$$

Then, entities for a paper search query may include author names, paper titles, and the query itself.

Entity Document Retrieval For each mentioned entity $i \in \mathcal{E}_q$, we then use an off-the-shelf text embedding model to retrieve its associated textual document d_i from $\mathcal{D}$. As shown in Figure 2, author documents contain information of paper and citation counts while paper documents contain abstract and publication information such date and venue.

KG Relation Propagation Based on the semi-structured knowledge base, we then link each document to its corresponding entity node $v_i \in \mathcal{V}$ on KG. For each node v_i, we extract based on the KG relations in $\mathcal{G}$ its h-hop neighbors. We denote the set of these neighbor nodes as $\mathcal{N}_i$ and their relations to v_i as the set $\mathcal{R}_i$, e.g., author writes paper, paper cites paper. Similarly, we link each neighbor node $v_j \in \mathcal{N}_i$ to its corresponding textual document d_j.

Document-based Relation Filtering For a dense KG, a node might have a large amount of neighbors including nodes that are irrelevant to the query. Existing KG-based methods (Yasunaga et al., 2021;

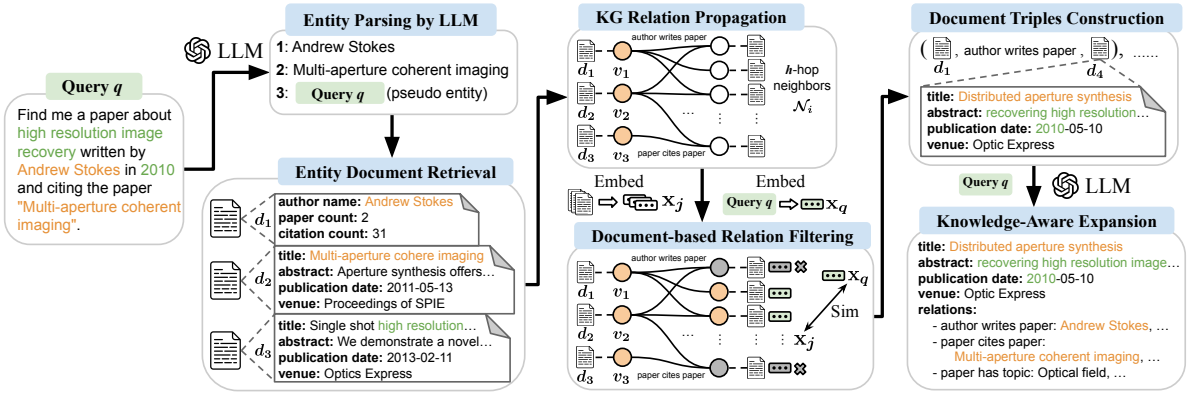

Figure 2: Overview of our knowledge-aware query expansion framework illustrated with an example academic paper search query with textual and relational requirements.

Zhang et al., 2022; Taunk et al., 2023) filter out irrelevant relations by scoring the relevance between nodes and queries based on entity names. Such entity-based approach, however, overlooks the rich textual details of entities. For example, an entity name in KG for paper search in Figure 2 is simply a paper title, while further details of the paper, such as the abstract content "high resolution image recovery" and publication information "2010", are often not directly available in KG, despite being frequently queried by users.

To this end, we leverage the associated document texts as rich representations of KG nodes and use document-based relation filtering to get query-focused relations. Specifically, given the set of neighbor nodes $\mathcal{N}_i$, instead of using simply the entity names, we embed the textual document d_j for each neighbor node with a text embedding model as $\mathbf{x}_j = \text{Embed}(d_j)$ and we also embed the original query q using the same embedding model $\mathbf{x}_q = \text{Embed}(q)$. Then, we compute the semantic similarity of each node v_j with the query q and score them as

$$s_{j,q} = \text{Sim}(\mathbf{x}_j, \mathbf{x}_q). \tag{4}$$

which reflects more accurately the relevance between query and the neighbor node utilizing richer textual details besides entity name. Based on the similarity scores, we select the top-k scored nodes as query-focused neighbors

$$\mathcal{N}_{i,q} = \{v_j \in \mathcal{N}_i \mid s_{j,q} \in \text{TopK}(s_q)\}, \tag{5}$$

and derive corresponding query-focused relations $\mathcal{R}_{i,q} \subseteq \mathcal{R}_i$. Since our document-based relation filtering uses an off-the-shelf text embedding model, it does not requires any re-training as most graph neural networks do when new nodes are added to the KG, showing the scalability of our method.

Document Triples Construction With our filtered neighbors nodes and relations, instead of constructing entity-based knowledge triples like Sun et al. (2024) and Luo et al. (2024), we further leverage the rich textual information to construct a document-based knowledge triples as in Figure 2

$$\mathcal{T}_{i,q} = \{(d_i, r_{i,j}, d_j) \mid v_j \in \mathcal{N}_{i,q}, r_{i,j} \in \mathcal{R}_{i,q}\}, \tag{6}$$

where $r_{i,j}$ denotes the relation on KG from node v_i to node v_j, e.g., paper cites paper, while d_i and d_j are the document texts associated with v_i and v_j containing details of each node, e.g., paper abstract, venue, and publication date. Such document triples not only provide rich textual details but also preserve the structured relational knowledge from KG to enhance the information accuracy.

Knowledge-Aware Expansion At the last step, we transform our document triples $\mathcal{T}_q$ into texts together with the original query q as LLM inputs. Leveraging its strong textual reasoning ability, we prompt the LLM to extract useful information from $\mathcal{T}_q$ and generate query expansions that help answer the query q as

$$\mathcal{Q}_e = \text{LLM}(q, \mathcal{T}_q), \tag{7}$$

where we follow Shen et al. (2024) and Chen et al. (2024) to sample n responses through a single LLM inference and concatenate them as the final expansion appended to the original query. The expanded query q' as defined in Equation 2 is then utilized for the final embedding-based document retrieval.

Throughout the query expansion, we leverage collaboratively textual documents and KG relations to achieve our **K**nowledge-**A**ware **R**etrieval (KAR). Since our method is zero-shot, it can be applied with various off-the-shelf LLMs and text embedding models. Besides, since our method utilizes

	#entities	#text tokens	#relations	avg. degree
AMAZON	1,035,542	592,067,882	9,443,802	18.2
MAG	1,872,968	212,602,571	39,802,116	43.5
PRIME	129,375	31,844,769	8,100,498	125.2

Table 1: Statistics of textual and relational semi-structured retrieval datasets in STaRK benchmark.

document texts for KG node representations and thus requires no additional model finetuning, it is scalable and flexible as new documents are added to the knowledge base.

5 Experimental Setup

5.1 Datasets and Metrics

We evaluate our method on three textual and relational semi-structured retrieval datasets from the STaRK benchmark (Wu et al., 2024b):

- **AMAZON**: a product search dataset where textual documents for 1.0M entities are collected from Amazon reviews (He and McAuley, 2016) and Q&A records (McAuley et al., 2015) and 9.4M KG relations include products viewed or purchased together, brands and colors.
- **MAG**: an academic paper search dataset based on obgn-papers100M (Hu et al., 2020) and MAG (Wang et al., 2020), where textual documents of 1.9M entities include paper title, abstract, and publication details and 39.8M KG relations include citation and authorship information.
- **PRIME**: a precision medicine inquiry dataset where textual documents are collected from multiple sources for about 129K entities such as disease, drug, protein and gene, and 8.1M KG relations are from PrimeKG (Chandak et al., 2023).

The statistics of datasets are shown in Table 1, where AMAZON data has richer textual information while MAG and PRIME have denser relations. We use the official test sets of synthetic queries and leave-out sets of human-generated queries in the STaRK benchmark as well as the following evaluation metrics: **Hit@1**, **Hit@5**, Recall@20 (**R@20**), and Mean Reciprocal Rank (**MRR**). We present further ablation results on human-generated queries to showcase the generalizability of our method for handling real-world queries.

5.2 Baselines & Variants

We compare our KAR method with the following baselines and ablated variants in *Zero-Shot* setting:

- **Base**: retrieving based on the original query.
- **PRF**: the classic pseudo relevance feedback (Lv and Zhai, 2010; Li et al., 2022) approach expanding the query with top-n initial retrieval results.
- **HyDE** (Gao et al., 2023): generating expansions directly with an LLM based on the original query.
- **RAR** (Shen et al., 2024): a retrieval-augmented retrieval approach using top-n initially retrieved documents as contexts for generating query expansions with an LLM.
- **AGR** (Chen et al., 2024): a recent method using a multi-step framework to analyze, generate, and then refine based on top-n initial retrievals for expansion optimization with an LLM.
- **KAR$_{\text{w/o KG}}$**: an ablated variant of our proposed KAR method without access to KG relations and thus generates expansions based solely on textual documents of retrieved entities as in Equation 3.
- **KAR$_{\text{w/o DRF}}$**: an ablated variant of our proposed KAR method without **D**ocument-based **R**elation **F**iltering (DRF). Instead, it conducts entity-based relation filtering as in Yasunaga et al. (2021) and Zhang et al. (2022) using entity names.

For more comprehensive comparisons, we report results of some *Supervised* baselines from the STaRK benchmark, including **DPR** (Karpukhin et al., 2020) as a representative dense retrieval method, **QAGNN** (Yasunaga et al., 2021) as a representative language model embedding-augmented graph neural network method, and the state-of-the-art LLM retrieval agent **AvaTaR** (Wu et al., 2024a).

5.3 Implementation Details

For all LLM-based query expansion methods, we use Azure OpenAI API for GPT-4o (2024-02-01) as the backbone LLM in our main experiments. We also present additional results using LLaMA-3.1-8B-Instruct (Dubey et al., 2024) as backbone LLM in Section 6.5. Following Jia et al. (2024) and Shen et al. (2024), we use the dense embeddings from OpenAI `text-embedding-ada-002` model for all query expansion methods in our main experiments, employing the dot product for similarity calculation as well as document retrieval. We also show additional results of sparse retrieval using BM25 (Robertson et al., 2009) as retriever in Section 6.5. We truncate the input when its length exceeds the

Method	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
Supervised Settings												
DPR	15.29	47.93	44.49	30.20	10.51	35.23	42.11	21.34	4.46	21.85	30.13	12.38
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
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
Zero-Shot Settings												
Base	39.16	62.73	53.29	50.35	29.08	49.61	48.36	38.62	12.63	31.49	36.00	21.41
PRF	40.07	60.66	51.24	49.79	29.04	47.65	46.69	37.90	12.46	28.63	33.04	20.06
HyDE	40.31	64.43	53.71	51.42	29.98	50.10	50.02	39.58	16.85	37.59	43.55	26.56
RAR	51.52	66.63	54.63	58.73	39.02	52.87	50.87	45.74	22.53	40.84	44.50	30.93
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
KAR$_{\text{w/o KG}}$	43.54	60.29	51.83	51.80	31.14	46.75	46.86	38.88	18.03	36.27	42.00	26.84
KAR$_{\text{w/o DRF}}$	47.99	67.54	56.91	57.14	45.44	63.83	58.67	53.85	25.85	46.52	48.10	35.52
KAR	**54.20**	68.70	57.24	**61.29**	**50.47**	**65.37**	**60.28**	**57.51**	**30.35**	**49.30**	**50.81**	**39.22**

Table 2: Retrieval results on test sets of synthetic search queries.

context window of the backbone LLM or embedding model. All experiments run on an NVIDIA A100-SXM4-80G GPU. The prompts for all LLM-based methods are provided in Appendix A.

For hyperparameters, we set $n = 3$ for PRF and all other methods utilizing the top-n initial retrieval results following Chen et al. (2024) and Jia et al. (2024). Existing LLM-based query expansion methods (Gao et al., 2023; Shen et al., 2024; Chen et al., 2024) usually sample multiple expansions from a single LLM inference in Equation 7 to enhance the generation diversity and thus the coverage of relevant information. Thus, we follow them and set the default number of sampled expansion generations as the same $n = 3$ for a fair comparison with PRF. The influence of different number of sampled query expansions on retrieval accuracy is further discussed in Section 6.4. Regarding the KG parameters specifically in our KAR method, we set $h = 2$ for h-hop neighbors following Zhang et al. (2022) and Taunk et al. (2023) to avoid exponentially increasing number of neighbor nodes farther than 2-hop. We choose $k = 10$ to select top-k neighbors for query-focused relations. The ablation results with different k are presented in Section 6.3. We also discuss the relative latency of compared query expansion methods in Section 6.6.

6 Results

6.1 How does KAR perform in textual and relational semi-structured retrieval?

We show the results on test sets of synthetic queries in Table 2 and leave-out sets of human-generated queries in Table 3, from which we observe that our KAR method achieves consistently the best or second-best performance on all metrics, validating its effectiveness for textual and relational retrieval and its generalizability to real-world scenarios.

For query expansion baselines, we find that simply using initial retrieval results as expansions, i.e., PRF, has little or even negative impact on final retrieval accuracy as low-quality initial retrievals can introduce noise for final retrievals. Meanwhile, HyDE employs an LLM to generate query expansions directly with its intrinsic knowledge. However, without grounded textual knowledge from the document corpus, HyDE only improves retrieval performance marginally. For more advanced LLM-based methods, i.e., RAR and AGR, we observe that augmenting LLM with initial retrievals as contexts before expansion and utilizing its self-refinement abilities can indeed improve expansion quality and thus retrieval accuracy. However, the lack of relational knowledge can still lead to incorrect document relations limiting their performance on textual and relational semi-structured retrieval.

For supervised baselines, according to Wu et al. (2024b), training challenges of encoding both textual and relational information as texts for dense retrievers and computational demands for graph neural network in QAGNN lead to significant performance gaps. While LLM-based agent AvaTaR shows promising results after being optimized on training data, the cost and efficiency remain challenging with a high number of LLM inferences.

Moreover, compared to MAG and PRIME, we observe on AMAZON dataset higher general performance of all methods and smaller performance gaps between KAR and baselines, e.g., AvaTaR outperforms KAR on Hit@5 and R@20 metrics as shown in Table 2. The observation aligns well with the dataset characteristics in Table 1 that AMAZON has richer textual information which can be

Method	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
Base	39.50	64.20	35.46	52.65	28.57	42.86	36.40	35.95	22.02	41.28	43.98	30.63
PRF	43.21	64.20	30.19	53.53	29.76	41.67	32.91	35.66	24.77	36.70	40.65	30.35
HyDE	45.68	72.84	39.25	57.56	29.76	44.05	37.84	35.51	24.77	42.20	47.70	33.65
RAR	55.56	71.60	36.15	62.15	38.10	45.24	35.19	42.04	31.19	43.12	49.01	37.72
AGR	55.56	71.60	37.27	63.54	33.33	44.05	37.23	38.95	32.11	49.54	49.65	39.27
KAR$_{\text{w/o KG}}$	49.38	67.90	33.94	57.77	30.95	40.48	30.95	35.16	29.36	47.71	53.52	37.80
KAR$_{\text{w/o DRF}}$	56.79	**76.54**	39.95	65.72	41.67	55.95	44.54	48.99	34.86	56.88	56.21	44.51
KAR	**61.73**	72.84	**40.62**	**66.32**	**51.20**	**58.33**	**46.60**	**54.52**	**44.95**	**60.55**	**59.90**	**51.85**

Table 3: Retrieval results on leave-out sets of human-generated search queries.

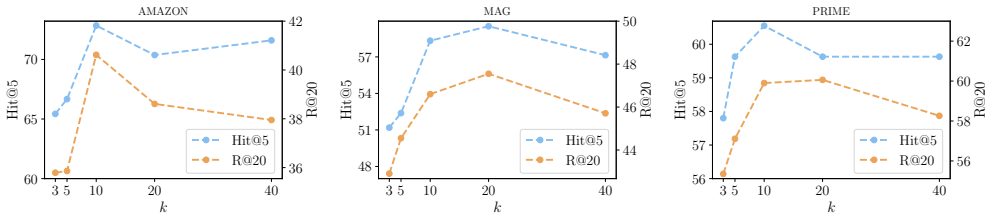

Figure 3: Influence of different values of k for filtered top-k neighbors in KAR.

handled better by LLMs while denser relational structures in MAG and PRIME pose challenges for LLMs to generate high quality expansions.

6.2 Are KG and document-based relation filtering (DRF) really effective?

In both Table 2 and 3, we show results of two ablated variants of our method: KAR$_{\text{w/o KG}}$ which has no access to relational knowledge, and KAR$_{\text{w/o DRF}}$ which conducts entity-based relation filtering similarly as in Yasunaga et al. (2021) and Zhang et al. (2022). KAR consistently outperforms these two variants except the Hit@5 metric on AMAZON in Table 3. The results shows that document texts and relations are both necessary and effective in enhancing the retrieval accuracy especially with denser relation structures and they contribute collaboratively to KAR. We also find that KAR$_{\text{w/o DRF}}$ achieves generally better performance than KAR$_{\text{w/o KG}}$. We attribute this result to the fact that LLMs' intrinsic knowledge can mitigate the textual semantic gap between queries and documents to some extent while they lack more structured relational knowledge that should be derived from the KG.

6.3 How does the number of filtered top-k neighbors affect KAR?

To further study the effectiveness of incorporating KG relations, we show retrieval results of KAR with varying $k \in [3, 5, 10, 20, 40]$ for filtered top-k neighbors in Figure 3. From the results, we find that initially including more query-focused neighbors based on textual documents can indeed improve retrieval accuracy as more useful document relations are covered. However, marginal improvement diminishes as k gets larger, and we observe a decrease in retrieval accuracy when increasing k to 40, which suggests that irrelevant neighbors have been included, introducing noisy document relations affecting LLM's query expansion quality. Nevertheless, our KAR method performs competitively well across different choices of k.

6.4 Does the number of sampled query expansions n affect retrieval accuracy?

Existing LLM-based methods (Gao et al., 2023; Shen et al., 2024; Chen et al., 2024) often sample multiple expansions from a single LLM inference to enhance generation diversity and coverage. To study its influence on textual and relational retrieval, we show in Figure 4 the retrieval accuracy of LLM-based methods with a varying number of sampled expansions $n \in [1, 3, 5, 7]$. From the results, we observe that only on AMAZON do they exhibit a slightly increasing trend, while on MAG and PRIME, the number of sampled expansions does not have an obvious impact on retrieval accuracy. We attribute this to denser relations in MAG and PRIME, where more sampled expansions from LLMs do not help in identifying structured document relations and thus cannot improve retrievals.

6.5 Can KAR work with other retrievers and backbone LLMs?

To further demonstrate the flexibility and scalability of KAR, we show in Table 4 retrieval results

Method	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
Base	34.57	55.56	22.78	44.31	32.14	41.67	29.32	36.88	23.85	43.12	39.73	31.19
PRF	38.27	56.79	22.19	46.45	29.76	50.00	37.83	38.08	25.68	40.37	41.96	32.66
HyDE	36.63	58.44	23.39	46.29	32.14	**50.40**	**39.72**	39.84	28.44	44.65	44.40	35.79
RAR	40.74	59.26	23.39	49.09	34.23	50.00	38.41	40.90	29.82	45.18	45.05	37.02
AGR	43.46	60.25	**24.12**	50.90	34.76	48.57	36.90	40.59	30.64	45.50	44.94	37.50
KAR	**45.06**	**61.32**	24.10	**52.24**	**36.51**	50.20	38.69	**42.41**	**32.26**	**47.25**	**46.06**	**39.14**

Table 4: Results with BM25 as retriever on human-generated queries.

Method	AMAZON				MAG				PRIME			
	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR	Hit@1	Hit@5	R@20	MRR
Base	39.50	64.20	35.46	52.65	28.57	42.86	36.40	35.95	22.02	41.28	43.98	30.63
PRF	43.21	64.20	30.19	53.53	29.76	41.67	32.91	35.66	24.77	36.70	40.65	30.35
HyDE	43.21	65.43	36.11	53.92	22.62	38.10	29.78	29.23	21.10	39.45	42.61	29.95
RAR	50.62	62.96	33.67	57.84	33.33	41.67	32.01	37.37	28.44	44.95	50.12	36.13
AGR	51.85	70.37	35.70	59.91	30.95	39.29	32.11	35.20	23.85	44.95	50.00	34.43
KAR	51.85	**71.60**	**37.78**	**60.70**	**42.86**	**54.76**	**41.36**	**47.31**	**41.28**	**55.96**	**56.45**	**48.12**

Table 5: Retrieval results with LLaMA-3.1-8B-Instruct as backbone LLM on human-generated queries.

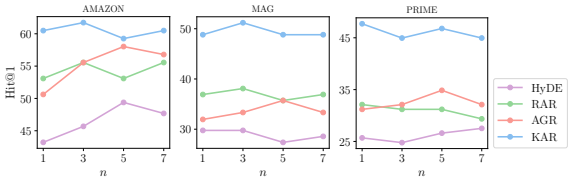

Figure 4: Influence of sampled query expansions n.

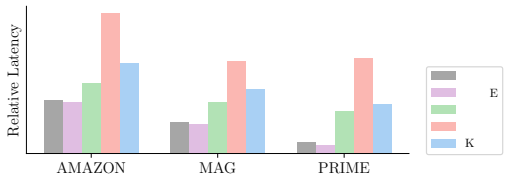

Figure 5: Latency comparison of query expansions.

of all compared query expansion methods using BM25 as the sparse retriever in replacement of dense embedding-based retrieval, and in Table 5 results with LLaMA-3.1-8B-Instruct as the backbone LLM for expansion generations. We use the same prompts as in Appendix A and we observe similar performance trends across different methods, with KAR consistently being the best or the second-best. We also find that sparse retrievals based on BM25 have lower performance than dense embedding-based retrievals when augmented with query expansions, which indicates the complexity of textual and relational semi-structured retrieval.

6.6 How does KAR compare to other methods in terms of retrieval latency?

Since the latency of individual method varies based on different implementations and API versions, we keep them consistent for all methods as specified in Section 5.3 and report results as in Figure 5 for relative comparisons. We observe that PRF and HyDE achieve the lowest latency as they only introduce one additional retrieval or one LLM inference before final retrieval. RAR uses initial retrieval results as contexts for LLM inference, resulting in higher latency due to the additional retrieval as well as the increased textual inputs. AGR implements a multi-step refinement framework involving five LLM inferences, leading to about twice the latency of RAR and such observation is also suggested by Chen et al. (2024). Instead of introducing extra LLM inferences, our KAR method employs KG to provide structured relational knowledge, allowing for fast inference, achieving considerable performance improvements while only introducing only a small amount of additional latency.

7 Conclusion

In this paper, we develop a knowledge-aware query expansion framework for textual and relational retrieval, utilizing KG relations between textual documents to enhance LLMs' query expansion generation. Leveraging collaboratively textual and relational knowledge, we filter query-focused relations with document texts as rich KG node representations for our knowledge-aware retrieval. Experiments on three semi-structured retrieval datasets of diverse domains demonstrate the advantages of our method compared against state-of-the-art query expansion methods, and showcase its applicability to handle real-world complex search queries.

Limitations

Similar to all existing LLM-based query expansion methods, one limitation of our KAR method is the retrieval efficiency as discussed in Section 6.6. While we have optimized our framework to only incorporate two LLM inferences per search query, the latency for API calls may also be influenced by varying server load. Though we also show the effectiveness of KAR with LLaMA-3.1-8B-Instruct in Section 6.5 for local LLM inference, the computation constraints and deployment costs are additional important factors to be taken into consideration for practical applications. Therefore, future works may further explore the acceleration and cost optimization of LLM inference, e.g., parallel inference, for more efficient query expansions.

References

Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735.

Derian Boer, Fabian Koch, and Stefan Kramer. 2024. Harnessing the power of semi-structured knowledge and llms with triplet-based prefiltering for question answering. *arXiv preprint arXiv:2409.00861*.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6101–6119, Dublin, Ireland. Association for Computational Linguistics.

Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67.

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 and virtual meeting. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. Grag: Graph retrieval-augmented generation. *arXiv preprint arXiv:2405.16506*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. 2024. MILL: Mutual verification with large language models for zero-shot query expansion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2498–2518, Mexico City, Mexico. Association for Computational Linguistics.

Xinke Jiang, Ruizhe Zhang, Yongxin Xu, Rihong Qiu, Yue Fang, Zhiyuan Wang, Jinyi Tang, Hongxin Ding, Xu Chu, Junfeng Zhao, et al. 2024. Hykge: A hypothesis knowledge graph enhanced framework for accurate and reliable medical llms responses. *arXiv preprint arXiv:2312.15883*.

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.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401, St. Julian's, Malta. Association for Computational Linguistics.

Hang Li, Shengyao Zhuang, Ahmed Mourad, Xueguang Ma, Jimmy Lin, and Guido Zuccon. 2022. Improving query representations for dense retrieval with pseudo relevance feedback: A reproducibility study. In *European Conference on Information Retrieval*, pages 599–612. Springer.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Linhao Luo, Yuan-Fang Li, Reza Haf, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 579–586.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Liana Patel, Siddharth Jha, Carlos Guestrin, and Matei Zaharia. 2024. Lotus: Enabling semantic queries with llms over tables of unstructured and structured data. *arXiv preprint arXiv:2407.11418*.

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

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15933–15946, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Giuseppe Spillo, Cataldo Musto, Marco Polignano, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2023. Combining graph neural networks and sentence encoders for knowledge-aware recommendations. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, pages 1–12.

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.

Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. 2023. Grapeqa: Graph augmentation and pruning to enhance question-answering. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 1138–1144, New York, NY, USA. Association for Computing Machinery.

Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, et al. 2024. Learning to plan for retrieval-augmented large language models from knowledge graphs. *arXiv preprint arXiv:2406.14282*.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

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-assisted knowledge retrieval. In *NeurIPS*.

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*.

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

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.

Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. 2003. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th international conference on World Wide Web*, pages 11–18.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

A Prompts for LLM-based Methods

In this section, we provide the prompts for all LLM-based query expansion methods in our experiments. The prompts for HyDE and RAR are presented in Table 6, and Table 7 shows the prompts for different modules in AGR. The prompts for our KAR method are provided in Table 8. While we generally follow the original prompts for all compared methods (Gao et al., 2023; Shen et al., 2024; Chen et al., 2024), slight adjustments are made to adapt these methods on the evaluated textual and relational retrieval tasks. For example, we follow Wu et al. (2024a) to provide LLMs with the structure information of documents in each dataset in the STaRK benchmark, which helps them to generate better formatted query expansions. The document structures of each dataset utilized in the prompts are provided in Table 9. For more details of the datasets, please refer to Wu et al. (2024b).

Method	Prompt
HyDE	"""Given the document structures: {doc_struct}, write a document that answers the following user query. Return the document only without any additional text. Query: {query} Document: """
RAR	"""Given the document structures: {doc_struct} and initially retrieved documents: {PRF_doc_1} {PRF_doc_2} {PRF_doc_3} write a document that answers the following user query. Return the document only without any additional text. Query: {query} Document: """

Table 6: Prompts for HyDE and RAR.

AGR	**Prompt**
Extract	"""Given the following user query, write a list of keywords. Return the keywords only without any additional text. Query: {query} Keywords: """
Analyze	"""Given the following user query and extracted keywords: {extracted_keywords}, do not attempt to explain or answer the question, just provide the query analysis: Query: {query} Analysis: """
Generate[1]	"""Given the document structures: {doc_struct} and the query analysis: {query_analysis}, write a document that answers the following user query. Return the document only without any additional text. Query: {query} Document: """
Generate[2]	"""Given the document structures: {doc_struct} and initially retrieved documents: {AGR_retrieved_doc_1} {AGR_retrieved_doc_2} ... {AGR_retrieved_doc_9} write a document that answers the following user query. Return the document only without any additional text. Query: {query} Document: """
Refine	"""Given the candidate documents: {AGR_generated_doc_1} {AGR_generated_doc_2} {AGR_generated_doc_3} evaluate the accuracy and reliability of each candidate document. Identify any misinformation or incorrect facts in the answers. Then write a correct document that best answers the following user query. Return the document only without any additional text. Query: {query} Document: """

Table 7: Prompts for different modules in AGR.

KAR	**Prompt**
Parse	"""Given the document structures: {doc_struct}, identify named entities in the following user query. Follow the document structures, write a document for each entity in the format: {document type: {document attributes}}. Query: {query} Documents: """
Generate	"""Given the document structures: {doc_struct} and retrieved textual and relational documents: {KAR_document_triples} extract useful information that help answer the following user query. Then, write a document that answers the following user query. Return the document only without any additional text. Query: {query} Document: """

Table 8: Prompts for different modules in KAR.

Dataset	**Document Structures**

AMAZON	{
	"product": ["title", "brand", "description", "features", "reviews", "Q&A"],
	"brand": ["brand_name"],
	"category": ["category_name"],
	"color": ["color_name"]
	}

MAG	{
	"paper": ["title", "abstract", "publication date", "venue"],
	"author": ["name"],
	"institution": ["name"],
	"field_of_study": ["name"]
	}

PRIME	{
	"disease": ["id", "type", "name", "source", "details"],
	"gene/protein": ["id", "type", "name", "source", "details"],
	"molecular_function": ["id", "type", "name" "source"],
	"drug": ["id", "type", "name", "source", "details"],
	"pathway": ["id", "type", "name", "source", "details"],
	"anatomy": ["id", "type", "name", "source"],
	"biological_process": ["id", "type", "name", "source"],
	"cellular_component": ["id", "type", "name" "source"],
	"exposure": ["id", "type", "name", "source"]
	}

Table 9: Document structures of the three datasets.