HubLink: A Novel Question Answering Retrieval Approach over Knowledge Graphs

Angelika Kaplan[1], Jan Keim[1], Marco Schneider[1] and Ralf Reussner[1]

[1]Karlsruhe Institute of Technology (KIT), Kaiserstraße 12, 76131 Karlsruhe, Germany

Abstract

The rapid growth of scholarly literature poses challenges for efficient and effective information retrieval. Existing Question Answering over Knowledge Graphs (KGQA) systems, particularly those relying on Semantic Parsing, struggle with schema dependency and required training data. In this paper, we introduce HubLink, a schema-agnostic, training-free KGQA approach leveraging pre-trained Large Language Models to enhance scholarly search with semantic aspects. HubLink structures (research) knowledge graphs into conceptual hubs, enabling source-aware inference for literature. For evaluation, we use the Open Research Knowledge Graph as the underlying knowledge base and utilize a dataset from software architecture research to populate the graph. The empirical results show that our approach HubLink outperforms three state-of-the-art baselines, especially for complex queries, marking a major advancement in scholarly KGQA. In future work, we aim to explore more advanced techniques to improve the final answer generation.

Keywords

Research Knowledge Graphs (RKG), Question Answering over Knowledge Graphs (KGQA), Retrieval-Augmented Generation (RAG)

1. Introduction

Despite advances in digital publishing, scholarly communication remains fundamentally document-centric, with scientific knowledge fragmented across isolated articles [1, 2]. This prevents researchers from effectively interlinking related findings, methodologies, and datasets across publications [2]. The resulting isolation creates major barriers to knowledge discovery, particularly as the exponential growth of scientific literature makes comprehensive literature review increasingly infeasible [1, 2]. Furthermore, traditional keyword-based search methods in this context are hindered by lexical variability, including synonyms, abbreviations, and misspellings, further complicating information retrieval [3].

Recent progress in Large Language Models (LLMs) offers new opportunities for natural language-based scholarly search. In particular, LLMs show potential for enabling question answering (QA) interfaces that allow to query large bodies of scientific content. However, relying solely on the internal knowledge of pre-trained LLMs is problematic, as it often leads to shallow, unverifiable, or hallucinated responses [4]. Retrieval-Augmented Generation (RAG) has emerged as a promising approach to mitigate these issues by enriching LLM responses with context retrieved from an external knowledge base [5]. However, current naive RAG systems, i.e., based on embeddings and vector stores, suffer from limited retrieval precision and fail to synthesize information coherently across multiple sources [6].

To address these limitations, structured representations of scholarly knowledge, specifically Research Knowledge Graphs (RKGs) such as the Open Research Knowledge Graph (ORKG) [1, 2], have been proposed as a means to transition from document-centric to knowledge-centric scholarly communication. RKGs represent entities and relationships extracted from scholarly content in a machine-readable format, enabling more expressive and interconnected queries.

RAGE-KG 2025: The Second International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2025, November 2–6, 2025, Nara, Japan

angelika.kaplan@kit.edu (A. Kaplan); jan.keim@kit.edu (J. Keim); marco.schneider@student.kit.edu (M. Schneider); ralf.reussner@kit.edu (R. Reussner)

https://dsis.kastel.kit.edu/staff_angelika_kaplan.php (A. Kaplan); https://mcse.kastel.kit.edu/staff_jan_keim.php (J. Keim); https://dsis.kastel.kit.edu/staff_ralf_reussner.php (R. Reussner)

0009-0009-9101-5833 (A. Kaplan); 0000-0002-8899-7081 (J. Keim); 0000-0002-9308-6290 (R. Reussner)

While KGQA has achieved notable success in open-domain applications, its adaptation to scholarly knowledge remains largely unexplored. The few existing approaches in the scientific domain predominantly rely on semantic parsing (SP) methods, which require extensive training data and domain-specific query templates. These SP-based systems face critical limitations: they struggle to scale across diverse research domains, cannot adapt to the continuously evolving schemas of RKGs, and fail to capture the nuanced semantic relationships inherent in academic discourse. Moreover, their dependence on predefined patterns makes them ineffective for the complex and multifaceted queries typical of scholarly queries. Therefore, we aim to advance academic knowledge discovery by creating a schema-agnostic KGQA system for complex scholarly queries without requiring domain-specific training.

To address our objective, we ask the following research questions (RQs):

1. *How can the limitations of schema-dependent and training-based KGQA systems in scholarly domains be overcome by integrating RKGs with LLMs for schema-agnostic and provenance-aware retrieval?*
2. *What is the performance of such an approach when benchmarked against state-of-the-art approaches on scholarly datasets?*

To address our research questions and overcome the limitations of existing approaches, we propose HubLink, a novel training-free and schema-agnostic retrieval approach for scholarly KGQA that organizes Knowledge Graphs (KGs) into conceptual hubs to enable efficient, source-aware retrieval. To systematically evaluate KGQA systems in scholarly contexts, we introduce a comprehensive taxonomy for categorizing academic queries based on their complexity and information needs. Using this taxonomy, we evaluate HubLink on a literature search task in the software architecture (SWA) research domain utilizing the ORKG [1, 2] as KG. Our experimental results show that HubLink outperforms three state-of-the-art KGQA baselines, performing particularly better at complex, multi-hop queries that require information from multiple sources. These contributions advance the field toward more effective and scalable KGQA systems for academic knowledge discovery. The supplementary material associated with this paper is publicly available in our replication package [7].

2. HubLink: KGQA by Graph Decomposition

In this section, we present HubLink in response to our first research question, which addresses the challenge of overcoming schema dependencies while enabling provenance-aware retrieval. HubLink is our novel schema-agnostic, training-free KGQA approach that employs a GraphRAG-inspired pipeline comprising indexing, retrieval, and generation stages. The approach decomposes KGs into semantically coherent subgraph structures, so-called "hubs", during indexing, enabling source-aware retrieval with full traceability and provenance, which are essential requirements for scholarly literature search. The indexing phase identifies root entities, constructs hubs through structured graph traversal, and stores them in vector databases for efficient retrieval (Section 2.1). HubLink implements two retrieval strategies: *Graph Traversal Strategy* and *Direct Retrieval Strategy* (Section 2.2). The generation phase synthesizes answers while maintaining explicit source links to graph origins, ensuring both relevance and traceability (Section 2.3). This architecture delivers transparent KGQA with verifiable provenance, addressing the interpretability demands of academic and research contexts. An overview of our retrieval approach, HubLink, is depicted in Figure 1.

2.1. Indexing

HubLink's indexing process transforms KGs into structured sets of interlinked subgraphs (Hubs) that serve as primary retrieval units for Question Answering (QA). The process begins by selecting **start entities** as initial traversal points, then identifies **Hub Root** entities through graph traversal until reaching leaf nodes or other Hub Roots. Each Hub Root anchors **Hub Paths**, i.e., directed paths to terminal nodes. Hub Roots and their associated Paths collectively define Hubs.

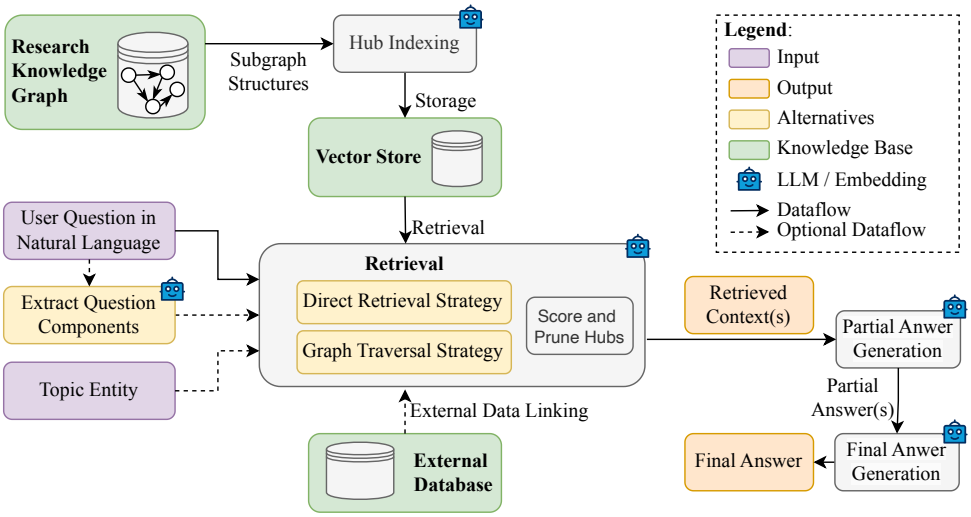

Figure 1: Overview of our approach HubLink, covering the phases of indexing, retrieval, and answer generation

Each Hub Path undergoes a multi-step transformation and indexing pipeline: (1) LLM generation of textual descriptions, (2) parsing into structured components, including RDF-like triples and entities, (3) embedding into a vector space via pre-trained models, and (4) storage in vector databases supporting Approximate Nearest Neighbor (ANN) search [8, 9].

The vectors include essential metadata (Hub Root identifiers & LLM-generated descriptions), enabling traceable retrieval during queries. The indexing process is recursively applied to Hub Path endpoints until it reaches leaf nodes or the maximum traversal depth, ensuring controlled graph coverage.

Since retrieval relies entirely on indexed data, maintaining consistency between knowledge graphs and indices is critical. As such, the approach implements two complementary update strategies: *Fixed Update* for complete scheduled synchronization and *Dynamic Update*, where the (real-time) changes in the graph define the required updates, enabling consistent KGQA in dynamic graph environments.

2.2. Retrieval

HubLink's retrieval identifies relevant hubs and paths for generating partial answers used in final answer generation. Two strategies are supported: (a) Direct Retrieval Strategy for fast retrieval with potential accuracy reduction in locally scoped queries, and (b) Graph Traversal Strategy for higher precision in local queries at increased execution cost. Both strategies require question preprocessing: (i) computing full question embeddings via pre-trained models, (ii) extracting semantic components (entities, types, time expressions, constraints) via LLM for precise matching, e.g., "Which papers have been published by CEUR-WS.org in 2020?" decomposes to ['Publisher', 'CEUR-WS.org', '2020'], and (iii) individually embedding and aggregating extracted components. To support fine-grained retrieval for complex queries, the input includes the original question, semantic components, and vector representations.

The **Direct Retrieval Strategy** uses ANN search [8, 9] within the precomputed vector index (cf. Section 2.1) for efficient KG content retrieval without graph queries, emphasizing speed and simplicity. Following question preprocessing, the strategy iteratively queries the vector store to collect candidate hubs, avoiding duplicates and terminating early when no new results are found. Retrieved hubs undergo refinement to standardize path counts: excess paths are pruned while insufficient hubs receive additional paths via vector search based on hub root entities. Final hubs are scored, ranked, filtered for processing, and then generate intermediate responses. If one or more partial answers exist, they are synthesized into a coherent final answer (cf. Section 2.3).

Starting from Hub roots, the **Graph Traversal Strategy** explores the KG structure to identify relevant information, incorporating structural graph information through bidirectional path traversal.

The process initializes bidirectional search from the entry entity, preprocessing the input question as described above. Using these values, traversal proceeds iteratively up to a predefined maximum depth. At each level, the algorithm checks for available entities, identifies reachable hub candidates from the current entity set, and determines the next entities for exploration. Found hub candidates are ranked by relevance, with top-ranked hubs used to generate partial and final answers (cf. Section 2.3). If no answers are produced, traversal continues to the next level.

HubLink's **Pruning** filters irrelevant hubs before answer generation, increasing efficiency and accuracy by evaluating candidate hubs based on their associated path relevance. Each hub receives a weighted relevance score computed by aggregating individual path scores, with higher-scoring paths receiving greater weight to ensure highly relevant information dominates the hub's final score. This weighting prioritizes hubs that contain highly informative paths while minimizing the impact of weaker paths. The approach ranks the scored hubs and keeps only those that exceed a predefined threshold. The goal is to produce semantically aligned hubs that support focused, accurate responses.

2.3. Answer Generation and Information Linking

Partial answer generation serves as HubLink's key intermediate step, building the foundation for final answer synthesis. This step utilizes KG information and supports the integration of external contextual data through linking procedures, enabling enhanced reasoning despite graph incompleteness or insufficient content. The generation takes a hub and processed question as input; for the graph traversal strategy, it includes a path connecting the hub to central topics, converted to natural language. The LLM receives the question, Hub Path descriptions, triples, and external knowledge, generating partial answers only when data sufficiently supports meaningful responses; otherwise, the Hub is skipped. This ensures context-aware, semantically grounded answers that are potentially enriched beyond the graph.

Final answer generation synthesizes partial answers into a single response. To achieve this, HubLink takes the user's question, the partial answers, and the paths they came from, and prompts an LLM to merge them. The prompt can be adjusted based on the task, e.g., in a literature search, the model is asked to insert citation tags into the answer to indicate where each claim came from. At the end of the answer, the approach adds a list of all the sources that were referenced. To make the results explainable and traceable, HubLink also returns a list of the actual triples (subject-predicate-object statements) from the KG that were used to generate the answer. It starts by sorting the retrieved paths by relevance, then collects all the triples they contain. These triples are passed, along with the question and final answer, to an LLM, which identifies only those triples that were actually relevant to the answer. This means that users not only get a clear and concise answer but also see which parts of the graph were used to create it, supporting transparency and helping build trust. This feature is especially useful in research contexts, where it is important to know an information's origin.

3. Evaluation

In this section, we evaluate HubLink in terms of performance benchmarking, addressing our second research question. To systematically assess retrieval quality and answer alignment, we use the Goal, Question, Metrics (GQM) approach [10, 11]. We first describe our experimental setup (Section 3.1), then present results organized by evaluation goals (Section 3.2).

3.1. Setup

This section describes the experimental setup for evaluating HubLink against state-of-the-art baselines.

3.1.1. Evaluation Metrics

In RAG systems, retrieval quality is fundamental to overall performance, as irrelevant or missing documents directly impact the quality of generated answers.

We employ various common information retrieval metrics to evaluate the effectiveness of document retrieval. For this, we use four rank-agnostic metrics. **Accuracy** provides an assessment of the ratio of correctly retrieved contexts compared to the total number of retrieved contexts. **Precision** quantifies the fraction of retrieved documents that are relevant, indicating the retrieval system's ability to avoid false positives. **Recall** measures the fraction of relevant documents that are successfully retrieved, capturing the system's completeness in finding information. The **F1 score** combines precision and recall into a single harmonic mean, providing a balanced measure that penalizes systems performing poorly on either metric. Additionally, we employ rank-aware metrics that consider the order of retrieved documents, as ranking quality is crucial for RAG systems that typically use only the top-k results. **Hits@K** measures the fraction of queries where at least one relevant document appears within the top-k retrieved results, providing insight into retrieval success at different cut-off points. **Mean Reciprocal Rank (MRR)** computes the average of the reciprocal ranks of the first relevant document for each query, emphasizing the importance of ranking relevant documents highly. **Mean Average Precision (MAP)** calculates the average precision across different recall levels for each query, offering a comprehensive view of ranking quality across all relevant documents. **Exact Match (EM)** quantifies the proportion of queries where the retrieved contexts exactly match the expected ground-truth contexts, providing a strict binary evaluation of retrieval correctness.

Beyond retrieval, we evaluate the quality of generated answers using established text generation metrics adapted for scientific QA contexts. **ROUGE** [12] calculates n-gram, word sequence, and longest common subsequence overlap between generated and reference texts, capturing lexical precision and content coverage essential for factual accuracy. **BLEU** [13] measures n-gram overlap between generated and reference answers, originally for machine translation but widely adopted for text generation. **BERTScore** [14] uses pre-trained transformer embeddings to compute semantic similarity rather than lexical overlap, suitable for scientific domains with paraphrasing and terminology variations. Using the RAGAs framework [15], we also use the following metrics: The **semantic similarity** compares the embeddings of the expected answer with the provided one, using OpenAI's *text-embedding-3-small* [16] and cosine similarity. Lastly, the **string similarity** uses basic string comparison metrics like Levenshtein, Jaro, Jaro-Winkler, and Hamming distance.

Complementing traditional metrics, we employ LLM-as-judge evaluation using LLMs to assess answer quality beyond lexical similarity [17, 18]. For this, we use the RAGAs framework [15] and the LLM *gpt-4o-mini* [19] by OpenAI. We use the following key metrics for scientific RAG evaluation from the RAGAs framework: **Faithfulness** ensures all answer claims are logically inferred from retrieved contexts, crucial for scientific QA requiring accuracy and verifiability. **Answer Relevancy** evaluates question-answer alignment, penalizing incomplete, verbose, or irrelevant responses. **Factual Correctness** measures alignment between generated answers and ground-truth references. **Instruction Following** evaluates whether generated answers comply with explicit formatting requirements in questions (e.g., ordering, aggregation, specific output structures).

When computing metrics across queries, multiple averaging approaches exist. We employ *macro-averaging*, which computes metrics per query then averages, treating each query equally. We use this averaging method for a balanced assessment across diverse query types and complexities, particularly important for evaluating KGQA systems on heterogeneous scholarly questions.

3.1.2. Knowledge Base Variants

To evaluate how knowledge representation affects retrieval performance, we constructed four RKG variants based on two orthogonal characteristics: path length and contribution granularity.

Path length determines the semantic depth of graph traversals. Long paths preserve rich semantic relationships through detailed intermediate nodes, enabling more expressive queries but requiring ex-

tensive traversal operations. Short paths collapse these relationships into direct connections, simplifying retrieval at the potential cost of semantic expressiveness.

Contribution granularity affects how research contributions are represented. Distinct contribution nodes create separate graph entities for each contribution within a paper, maintaining clear boundaries between different research outputs and their associated metadata. Cumulative contribution nodes consolidate all contributions from a paper into a single node, reducing graph complexity but potentially blending distinct research outputs.

Combining these characteristics yields four evaluation variants: Variant 1 **(GV1)** contains long paths with distinct contribution nodes (maximal semantic richness); Variant 2 **(GV2)** has long paths with cumulative contribution nodes (balanced semantic depth); Variant 3 **(GV3)** has short paths with distinct contribution nodes (simplified structure, preserved granularity); Variant 4 **(GV4)** contains short paths with cumulative contribution nodes (minimal complexity).

3.1.3. Question Taxonomy & Dataset

To structure our evaluation, we developed a comprehensive question taxonomy. This allows us to categorize questions and construct a question dataset that contains a fair mixture of different kinds of questions. For this paper, we focus on the categories based on retrieval operations required for answer generation. Drawing from KGQA dataset literature and scholarly research question frameworks [20, 21, 22, 23, 24, 25, 26, 27], we identified the following eight distinct question categories.

Basic questions enable direct answer retrieval without additional operations (e.g., "What is the definition of the Client-Server software architecture pattern?"). *Relationship* questions necessitate identifying connections or dependencies between information elements, such as causalities or correlations (e.g., "Which components in a client-server software architecture need to communicate with each other?"). *Negation* questions require detecting conditions that do not hold, based on explicit negation or absent information (e.g., "Which KGQA approach does not use training?"). *Aggregation* questions demand synthesizing multiple information pieces (e.g., "What is the average runtime of systems based on the client-server architecture?"). *Counting* questions involve enumerating relevant data points (e.g., "How many KGQA approaches were published between 2020 and 2024?"). *Superlative* questions require identifying extremes among multiple data points (e.g., "Which architecture pattern ensures lowest latency?"). *Ranking* questions necessitate ordering multiple data points according to specified criteria (e.g., "How do training-free KGQA approaches perform, sorted by F1 score?"). *Comparison* questions require evaluating attributes across two or more data points (e.g., "Is method A better than method B?").

We constructed our evaluation dataset using a curated, domain-specific dataset from the SWA research domain by Konersmann et al. [28] comprising 153 scientific publications from the European Conference on Software Architecture (ECSA) and the International Conference on Software Architecture (ICSA). Each publication is annotated w.r.t. multiple dimensions, including research object, evaluated property, evaluation method, paper class, bibliographic metadata, and others. An overview of the data schema is presented in [29]. We use this dataset to populate the ORKG according to each of the four knowledge base variants (see Section 3.1.2). Using these templates, we generated a QA dataset grounded in KG statements. The final dataset comprises 170 questions paired with ground-truth answers and corresponding KG statements required for answer generation, with variations adapted for each knowledge base variant.

3.1.4. Baselines

For the evaluation against state-of-the-art approaches, we established specific criteria for baseline selection. Given our focus on training-free scholarly KGQA, we required baseline approaches to be: (1) LLM-based KGQA systems, (2) training-free to ensure fair comparison with HubLink, and (3) accompanied by available and functional source code to enable reproducible evaluation. Our baseline selection process followed a systematic review of two comprehensive KGQA surveys by Pan et al. [30] and Peng et al. [31]. From this systematic review, we identified three approaches that met all our

criteria and represent different paradigms within training-free KGQA: Direct Fact Retrieval (DiFaR) [32], FiDeLiS [33], and MindMap [34]. Notably, all selected baselines are published at top-tier ACL venues (2023-2025), ensuring methodological rigor and community validation.

Direct Fact Retrieval (DiFaR) [32] indexes KG triples as embeddings, then uses ANN search to retrieve the closest triples for each query. Retrieved triples serve as context for answer generation, with an optional LLM-based reranking refinement step.

FiDeLiS [33] performs beam search from designated entry points using LLM-generated strategic plans that extract keywords and convert queries to declarative format. It iteratively retrieves and embeds relational paths, scoring them against keyword embeddings to maintain top-N candidates until either the termination criteria or the maximum path length is reached, then generates answers from the candidate paths.

MindMap [34] embeds all KG entities and uses LLM-driven entity extraction with ANN search to identify relevant entities from questions. It constructs evidence subgraphs, i.e., shortest paths between entities and 1-hop neighborhoods, then converts them to natural language descriptions via LLM to serve as context for answer generation.

3.1.5. Parameter Selection

HubLink, as well as the selected approaches, all require certain parameters to be selected. Due to practical constraints, including high costs and time requirements for LLM experiments, a fully-fledged parameter optimization process is infeasible. Thus, we employ a parameter selection process using the One-Factor-a-Time (OFAT) method [35] to optimize retrieval performance based on Recall and Hits@10 metrics. The following enumerates the individual parameters. Italic parameters served as the baseline for the OFAT method. In the evaluation, we use the parameters that have shown the most promising results. For parameter optimization, we utilized a reduced dataset of 44 diverse questions, following the same construction methodology as the full dataset, evaluated against GV1. Details on the individual results can be found in the replication package.

General parameters for all approaches: LLM models (Closed: *gpt-4o-mini* [19], gpt-4o [36], o3-mini [37]; Open: *Qwen2.5-14B* [38], Llama3.1-8B [39]), embedding models (*mxbai-embed-large* [40], text-embedding-3-large [16], granite-embedding [41]), question augmentation (*false*/true), and reranking (*false*/true).

HubLink-specific parameters: traversal strategy (*false*/true), extract question components (*false*/true), top paths to keep (*10*/20/30), number of hubs (*10*/20/30), filter output context (false/*true*), diversity ranking penalty (0/0.01/*0.05*/0.1), and path weight alpha (0/3/*5*/9).

Baseline parameters: DiFaR uses number of results (*30*/60/90/120/150) and distance metric (*cosine*/IP/L2); FiDeLiS employs top-k (*10*/20/30), top-n (*10*/20/30), and alpha (0.1/*0.3*/0.6); MindMap considers final paths to keep (*10*/20/30), shortest paths to keep (*10*/20/30), and neighbors to keep (*10*/20/30).

Besides the above parameters, we have different variants of **HubLink Configurations**: The **T** variant utilizes the graph traversal strategy of HubLink. This variant was also used in the parameter selection process. HubLink **D** instead uses the direct retrieval strategy. The variant **F** is a fast version that focuses on reduced runtime, employing the direct retrieval strategy, and limiting the number of hubs to 10 per question. The **O** variant adopts the T variant to use open models, i.e., the mxbai-embed-large embedding model and the LLM Qwen2.5-14B.

3.2. Evaluation Results

In this section, we present the empirical results of our comprehensive evaluation across two main goals: retrieval quality (Section 3.2.1) and answer generation quality (Section 3.2.2).

3.2.1. Evaluating Retrieval Quality

First, we evaluate the **retrieval quality (Evaluation Goal 1)**. To address the three following evaluation questions associated with this evaluation goal, we use the metrics Precision, Recall, F1-score, Hits@10,

Table 1
Retrieval performance of HubLink variants and baseline KGQA approaches on graph variant **GV1**

Approach	Recall	Precision	F1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	**0.754**	0.246	0.328	**0.512**	**0.299**	0.502	**0.298**
HubLink (D)	0.709	0.221	0.277	0.436	0.259	0.486	0.273
HubLink (F)	0.649	**0.278**	**0.344**	0.451	0.267	0.473	0.290
HubLink (O)	0.559	0.144	0.188	0.408	0.272	**0.526**	0.222
DiFaR	0.352	0.011	0.022	0.208	0.151	0.297	0.104
Mindmap	0.119	0.030	0.045	0.015	0.002	0.013	0.007
FiDeLiS	0.093	0.053	0.063	0.093	0.063	0.103	0.053

EM@10, MRR@10, and MAP@10.

Q1.1: What is the overall retrieval performance of the **different Hublink variants** in comparison to the baselines, and which HubLink variant performs best?

Table 1 shows retrieval performance of HubLink variants and baselines on GV1.

HubLink with graph traversal (T) achieves best overall performance, surpassing direct retrieval (D) which trades topic entity requirements for reduced effectiveness in complex graphs. HubLink (F), runtime-optimized with limited scope, achieves highest Precision/F1, illustrating hub count-performance trade-offs. The open variant (O) performs worst overall, though highest MRR@10 indicates effective ranking when retrieving correct triples. Comparing HubLink to the competing approaches, HubLink (T) achieves 0.754 Recall, doubling DiFaR's 0.352, far exceeding Mindmap/FiDeLiS 10%. Despite low Precision across models, HubLink (T)'s 0.246 shows improved relevance filtering. It leads in Hits@10 (0.512), MAP@10 (0.299), MRR@10 (0.502), though ranking limitations persist. HubLink variants outperform baselines in retrieval accuracy/contextual relevance, with graph traversal (T) proving most effective. Higher hub counts improve recall but reduce precision/ranking quality. Embedding-based methods offer clear advantages for scholarly KGQA despite filtering/ranking limitations.

Summary Q1.1: *HubLink substantially outperforms baseline KGQA approaches in both recall and precision, more than doubling the recall of the next best baseline. While ranking performance needs improvement, HubLink demonstrates superior retrieval of contextually relevant triples.*

Q1.2: Which performance influence do the different **required retrieval operations** have?

Table 2 shows the performance across the eight retrieval operations (cf. Section 3.1.3). The results are based on graph variant GV1. HubLink excels at Basic operations (highest Recall/Hits@10), indicating strong single-triple lookup performance. Precision/F1 scores remain lower than Recall, especially for Negation/Superlative operations, suggesting difficulty filtering relevant triples in complex reasoning. HubLink (T)/(O) excel at Basic operations; (D)/(F) perform better on Comparative/Relationship queries. Negation/Superlative operations consistently show reduced performance across variants.

Summary Q1.2: *Recall peaks for basic operations but declines with reasoning complexity. The retrievers struggle to distinguish relevant/irrelevant contexts, particularly for negation/superlative operations.*

Q1.3: How **robust to other graph schemas** is the best HubLink variant compared to the baselines?

Table 3 shows retrieval performance across graph variants. HubLink demonstrates superior robustness to graph variation compared to baselines. Shorter paths improve performance across all approaches, especially for Precision/ranking metrics. The baselines show steeper performance declines with increasing path length while HubLink maintains consistent superiority.

HubLink excels in multi-hop reasoning: as path lengths increase, baseline performance drops sharply while HubLink sustains high Recall and better Precision/ranking scores. Despite a slight decrease in precision with longer paths, HubLink remains most effective at retrieving deeply embedded information.

Summary Q1.3: *HubLink demonstrates greater robustness across diverse graph structures. While shorter paths improve all methods, HubLink notably outperforms baselines in multi-hop reasoning tasks.*

Table 2
Impact of the retrieval operation on the performance of HubLink and the KGQA baseline approaches

	Retrieval Operation	Recall	Precision	F1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	basic	**0.917**	**0.382**	**0.480**	**0.917**	**0.445**	0.490	**0.389**
	aggregation	0.810	0.209	0.285	0.497	0.225	0.347	0.240
	counting	0.840	0.275	0.372	0.644	0.357	0.526	0.340
	ranking	0.817	0.321	0.414	0.561	0.360	**0.576**	0.363
	comparative	0.742	0.262	0.366	0.456	0.320	0.560	0.296
	relationship	0.628	0.254	0.314	0.410	0.298	0.528	0.331
	negation	0.584	0.072	0.122	0.244	0.125	0.419	0.144
	superlative	0.656	0.129	0.193	0.319	0.207	0.540	0.237
DiFaR	basic	0.528	0.006	0.006	0.167	0.083	0.083	0.017
	aggregation	0.365	0.008	0.017	0.236	0.199	0.302	0.083
	counting	0.350	0.007	0.017	0.257	0.215	0.267	0.092
	ranking	0.164	0.005	0.013	0.111	0.089	0.190	0.058
	comparative	0.363	0.014	0.029	0.247	0.130	0.331	0.154
	relationship	0.380	0.017	0.032	0.270	0.239	0.545	0.196
	negation	0.364	0.013	0.031	0.219	0.107	0.308	0.131
	superlative	0.346	0.018	0.034	0.113	0.087	0.278	0.075
Mindmap	basic	0.278	0.020	0.037	0.056	0.006	0.006	0.006
	aggregation	0.115	0.031	0.046	0.004	0.000	0.004	0.004
	counting	0.182	0.041	0.065	0.004	0.000	0.004	0.004
	ranking	0.105	0.029	0.043	0.016	0.002	0.009	0.008
	comparative	0.026	0.010	0.014	0.005	0.005	0.042	0.004
	relationship	0.116	0.055	0.073	0.018	0.002	0.015	0.013
	negation	0.054	0.020	0.029	0.023	0.003	0.023	0.019
	superlative	0.079	0.030	0.041	0.000	0.000	0.000	0.000
FiDeLiS	basic	0.333	0.208	0.245	0.333	0.306	0.306	0.220
	aggregation	0.035	0.016	0.022	0.035	0.013	0.028	0.017
	counting	0.021	0.012	0.015	0.021	0.010	0.021	0.012
	ranking	0.102	0.037	0.051	0.101	0.038	0.072	0.029
	comparative	0.157	0.067	0.087	0.157	0.099	0.173	0.079
	relationship	0.052	0.044	0.048	0.053	0.033	0.121	0.046
	negation	0.011	0.011	0.011	0.010	0.010	0.062	0.006
	superlative	0.045	0.047	0.045	0.045	0.019	0.062	0.031

3.2.2. Evaluating Answer Alignment

Second, we evaluate the **answer generation quality (Evaluation Goal 2)** of the best HubLink variant (T) against the baselines on GV1. To answer the corresponding evaluation questions, we are using the metrics BLEU, ROUGE, Semantic/String Similarity, BERTScore, and the LLM-as-judge metrics Factual Correctness, Answer Relevancy, Instruction Following, and Faithfulness. Table 4 presents the results.

Q2.1: How semantically and factually consistent are the generated answers based on the retrieved triples of HubLink in comparison to the baselines?

HubLink achieves the highest factual correctness Recall (0.543), but substantially lower than retrieval Recall (0.754), indicating fact loss during generation. High ROUGE-1 Recall (0.757) may reflect lexical overlap rather than factual accuracy. DiFaR/Mindmap show better retrieval-generation alignment, suggesting more efficient fact retention. HubLink's BERTScore Recall (0.678) trails DiFaR's (0.702); lower precision across metrics indicates structural/lexical divergence from references. Despite higher factual correctness precision (0.301) than retrieval precision (0.246), HubLink lags behind DiFaR (0.290), suggesting inclusion of extraneous content. HubLink produces comprehensive but verbose answers, reducing factual focus.

Summary Q2.1: *HubLink shows limited fact integration and semantic consistency. Answers include*

Table 3
Evaluation results of the retrieval performance on different graph variants introduced in Section 3.1.2

	Graph	Recall	Precision	F1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	GV1	0.755	0.246	0.327	0.513	0.298	0.500	0.299
	GV2	0.759	0.276	0.352	0.518	0.333	0.522	0.324
	GV3	**0.812**	0.350	0.423	0.596	0.408	0.650	0.406
	GV4	0.804	**0.393**	**0.452**	**0.597**	**0.425**	**0.661**	**0.444**
DiFaR	GV1	0.352	0.011	0.022	0.207	0.150	0.295	0.104
	GV2	0.314	0.009	0.019	0.199	0.142	0.268	0.096
	GV3	0.523	**0.017**	**0.035**	0.302	**0.230**	0.442	0.154
	GV4	**0.528**	**0.017**	**0.035**	**0.304**	0.228	**0.449**	**0.158**
Mindmap	GV1	0.119	0.030	0.045	0.015	0.002	0.013	0.007
	GV2	0.093	0.025	0.037	0.008	0.001	0.006	0.005
	GV3	0.133	0.043	0.061	**0.030**	0.007	0.023	0.015
	GV4	**0.127**	**0.044**	**0.062**	**0.030**	**0.010**	**0.036**	**0.018**
FiDeLiS	GV1	0.092	0.052	0.063	0.092	0.062	0.103	0.053
	GV2	0.099	0.055	0.064	0.099	0.065	0.110	0.054
	GV3	0.259	0.114	0.139	0.259	0.150	0.240	0.112
	GV4	**0.276**	**0.121**	**0.142**	**0.276**	**0.156**	**0.248**	**0.117**

unrequested information and deviate structurally from references, evidenced by lower metric scores.

Q2.2: How relevant to the questions are HubLink's answers in comparison to the baselines?

HubLink achieves the highest answer relevancy (0.570), outperforming Mindmap (0.545), FiDeLiS (0.432), and DiFaR (0.203). However, 43% of answers lack optimal alignment. Notably, answer relevancy measures semantic fit, not factual accuracy or hallucinations.

Summary Q2.2: *HubLink leads in answer relevance, demonstrating strong semantic alignment with questions, although its absolute performance indicates room for improvement.*

Q2.3: How well does HubLink follow instructions in the question in comparison to the baselines?

Dataset questions often include explicit format instructions like ordering or aggregation. HubLink achieves the highest Instruction Following score (0.653), 68% above Mindmap/FiDeLiS (0.388) and double DiFaR (0.312). However, HubLink still fails to follow instructions in 34% of cases.

Summary Q2.3: *HubLink outperforms baselines in instruction following, but inconsistent adherence still requires improvement.*

Q2.4: How aligned are the answers of HubLink (T) and the baselines to the retrieved contexts?

Faithful answer generation requires that outputs are strictly based on the retrieved context, avoiding unsupported content. DiFaR achieves the highest Faithfulness score (0.645), while HubLink (0.445), Mindmap (0.396), and FiDeLiS (0.112) show progressively weaker grounding to retrieved contexts.

Summary Q2.4: *HubLink shows weaker contextual grounding than DiFaR. Improving faithfulness and reducing unsupported claims remain key areas for future work.*

4. Threats to Validity

This section analyzes potential threats to the validity of our experimental findings following the classification proposed by Konersmann et al. [28].

External validity refers to the generalizability of experimental results beyond the specific experimental context. One primary threat in our study is that the KGQA dataset questions may not fully capture the diverse information needs of researchers across different domains. To mitigate this risk, we systematically generated questions based on our structured taxonomy covering multiple query types and six representative use cases for scholarly literature search, ensuring broad coverage of typical research scenarios. Another threat concerns the use of experimental settings or tools that do not accurately

Table 4
Evaluation results for assessing the semantic and factual consistency of generated answers using the metrics Factual Correctness (FC), BERTScore (BERT), ROUGE, String Similarity (Str. Sim.), Semantic Similarity (Sem. Sim.), BLEU, Answer Relevancy, Instruction Following, and Faithfulness.

Approach	FC-Reca.	FC-Prec.	FC-F1	BERT-Reca.	BERT-Prec.	BERT-F1
HubLink (T)	**0.543**	**0.301**	**0.361**	0.678	0.515	0.580
DiFaR	0.387	0.290	0.321	**0.702**	0.588	**0.635**
Mindmap	0.203	0.212	0.184	0.652	0.625	0.633
FiDeLiS	0.131	0.201	0.154	0.516	**0.629**	0.562
Approach	ROUGE-Reca.	ROUGE-Prec.	ROUGE-F1	Str. Sim.	Sem. Sim.	BLEU
HubLink (T)	**0.757**	0.298	0.373	0.261	0.761	0.105
DiFaR	0.674	0.374	**0.448**	**0.338**	**0.772**	**0.160**
Mindmap	0.487	0.432	0.397	0.296	0.682	0.105
FiDeLiS	0.195	**0.503**	0.251	0.202	0.499	0.046

Approach	Answer Relevancy	Instruction Following	Faithfulness
HubLink (T)	**0.570**	**0.653**	0.445
DiFaR	0.203	0.312	**0.645**
Mindmap	0.545	0.388	0.396
FiDeLiS	0.432	0.388	0.112

reflect real-world usage conditions. To address this, we developed an evaluation framework using the RAG paradigm, which represents the current state of the art in QA systems. Additionally, we used standard evaluation metrics for RAG-based systems and their established formulas and implementations to ensure methodological rigor and comparability with related work.

Internal validity assesses whether observed effects stem from experimental manipulation rather than confounding factors. Potential threats include implementation variations between baselines and adaptation inconsistencies. To mitigate these, we used a custom framework integrating baseline retrievers implemented according to their original descriptions and source code. Adaptations were minimal to preserve original implementations, with all modifications documented for transparency.

Construct validity assesses whether experimental constructs are well-defined and properly operationalized. Given that our design is grounded in an established RAG evaluation framework and informed by recent surveys, the risk to construct validity is low. However, several potential threats remain: (1) testing a limited number of configurations per retriever may not capture the full performance space; (2) reliance on automated metrics without human evaluation may introduce mono-method bias; and (3) the OFAT design may overlook parameter interactions. To mitigate these risks, we used widely accepted metrics (e.g., precision, recall, RAGAS faithfulness, and relevance) and justified OFAT for its interpretability, applying systematic parameter selection.

Confirmability addresses the risk that experimental findings are influenced by researcher bias rather than being grounded in the underlying data. We acknowledge the potential for interpretive bias in our study: the selection of baseline systems may favor approaches similar to HubLink, the interpretation of ambiguous results could be influenced by our hypotheses, and the choice of evaluation scenarios might inadvertently advantage our approach. To mitigate these risks, we implemented multiple safeguards. First, we provide complete experimental artifacts, including raw results, processing scripts, and visualization code in our replication package. Second, we report both favorable and unfavorable results transparently, including cases where baselines outperform HubLink. Third, our baseline selection followed a systematic review of recent KGQA surveys and selected applicable retrievers.

Repeatability refers to the risk that experimental results may not be repeatable under the same or similar conditions. The primary threat in our study stems from the inherent non-determinism of LLMs, which may produce varying outputs across runs. While we anticipate that the relative performance of KGQA approaches (i.e., HubLink and the baselines) will exhibit consistent trends, absolute values may

fluctuate. To address this and to mitigate this risk, our framework is designed to capture, preserve, and replicate the complete experimental configuration.

5. Related Work

The following section reviews four key areas of related work in KGQA retrieval, each offering approaches conceptually relevant to our proposed HubLink approach.

Approaches KGQA on RKG: Recent scholarly QA approaches integrate LLMs with RKGs primarily through semantic parsing, translating natural language to formal queries (e.g., SPARQL) [42]. JarvisQA [43] and DBLP-QuAD [44] use annotated datasets and entity linking but suffer from schema dependence and limited scalability. KGMistral [45] and Taffa and Usbeck [46] employ template-based RAG, performing well in controlled settings but failing to generalize to unseen questions or dynamic KGs. Evaluation primarily uses the SciQA dataset, whose auto-generated nature and train-test schema overlap limit generalizability [47, 48], while semantic parser performance degrades on evolving KGs with unseen entities [49]. HubLink instead embeds subgraphs (hubs) rooted at specific nodes and transforms paths to text, enabling schema-agnostic adaptation without annotated training data.

LLM-Guided Stepwise KGQA Approaches: Recent training-free KGQA methods decompose tasks into iterative queries guided by pre-trained LLMs, differing from HubLink in how they structure and utilize KG information. The LLM acts as reasoning controller: Think-on-Graph (ToG) [50] uses beam search for multi-hop reasoning; Knowledge Solver (KSL) [51] models traversal as decision-making dialogue; Observation-Driven Agent (ODA) [52] operates in observe–act–reflect loops; GRAPH-COT [53] and StructGPT [54] call graph functions or linearize KG data; FiDeLiS [33] combines retrieval with deductive reasoning. Extensions include ToG-2 [55] with hybrid KG-text retrieval and Generate-on-Graph (GoG) [56] generating plausible triples for incomplete KGs. HubLink instead retrieves precomputed, embedded subgraphs (hubs), enabling reasoning over structured hubs rather than sequential LLM-driven exploration, thus incorporating transitive relations unreachable via stepwise traversal.

KGQA using Contextual Subgraph Construction: Some approaches construct subgraphs dynamically during retrieval by identifying question entities and expanding the KG to form localized subgraphs for LLM-based KGQA. Mindmap[34] prompts LLMs to assemble evidence subgraphs from paths and neighbors; KG-GPT [57] decomposes questions into triples before reasoning over linearized subgraphs; RoK [58] builds multi-hop paths via chain-of-thought reasoning, ranks them with PageRank, and forwards results to the LLM. Unlike these query-time dynamic approaches based on topology and proximity, HubLink precomputes semantically coherent subgraphs (hubs) offline, enabling fast retrieval with structured reasoning and origin tracking, features absent in dynamic methods.

Utilizing Vector Representations for KGQA: KGQA approaches using dense vector representations vary in embedding scope and training requirements. Direct embedding methods use pre-trained models without KG-specific training: DiFaR [32] encodes triples offline for nearest-neighbor retrieval but lacks structural context by focusing on isolated triples. HubLink instead embeds precomputed subgraphs (hubs), capturing richer semantics through textualized paths, triples, and entities. Training-based methods learn entity and relation vectors from KG structure: Pretrain-KGE [59] fine-tunes BERT on triples; KEPLER [60] combines Knowledge Graph Embedding (KGE) with language modeling; quaternion-based models [61] capture implicit semantics. However, all require substantial training data. HubLink uses general-purpose embeddings for textualized hubs, providing structured context without task-specific fine-tuning, ensuring portability across KGs.

6. Conclusion & Future Work

Current KGQA systems face critical limitations: schema dependence, training data requirements, and lack of standardized evaluation. This paper introduces HubLink, a schema-agnostic, training-free retrieval approach for scholarly KGQA. HubLink transforms RKGs into hubs, semantically coherent

knowledge clusters, enabling modular reasoning with full provenance tracking. The embedding-based retrieval uses LLM capabilities to eliminate schema dependencies and adapt to evolving KG structures.

Evaluation on ORKG with SWA data showed HubLink outperformed DiFaR, FiDeLiS, and MindMap across almost all metrics, excelling particularly in handling complex queries. The evaluation dataset based on our question taxonomy demonstrated HubLink's robustness. These results answer our second research question, demonstrating superior performance compared to state-of-the-art approaches on scholarly datasets.

This work demonstrates that HubLink's schema-agnostic, training-free approach with RKGs can help access scholarly knowledge, paving the way for more effective and scalable KGQA systems.

However, the evaluation also showed areas for improvement, including the following:

Improving Answer Generation. Our evaluation revealed limited alignment between retrieved contexts and generated answers. Future work should explore advanced prompt engineering and alternative synthesis methods beyond single-prompt generation.

Enhancing the Evaluation Scope. Current KGQA datasets operate at label-based granularity (Section 3.1.3). Extending HubLink to alternative text granularities could validate findings and enable HubLink's linking feature, which is ineffective with abstract label-based data.

Improving Relevancy Ranking. Results showed limitations in prioritizing relevant contexts. Future work should explore refined prompts or post-retrieval reranking mechanisms.

Augmenting Hub Content. Enriching hub indices with summaries or precomputed aggregations could enhance retrieval accuracy. This remains an open area for development and empirical validation.

Handling Numerical Constraints. While HubLink demonstrates temporal reasoning, complex numerical filtering remains underexplored and requires targeted evaluation.

Document-Based Retrieval Settings. Transitioning HubLink to document retrieval scenarios, where the KG supports source-aware retrieval, represents a promising direction. Preliminary work includes baseline implementations (e.g., LightRAG, GraphRAG) and structured knowledge extraction from scientific texts, though full evaluation remains pending.

Application to Alternative Graphs and Domains. The evaluation of HubLink was conducted on the ORKG as knowledge base within the SWA domain. Future work should assess the generalizability across diverse RDF-based RKGs and other domains.

Acknowledgments

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI 52/1 – project number 501930651, NFDIxCS, supported by funding from the pilot program Core Informatics at KIT (KiKIT) of the Helmholtz Association (HGF) and supported by KASTEL Security Research Labs.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, M. Vidal, Towards a knowledge graph for science, in: R. Akerkar, M. Ivanovic, S. Kim, Y. Manolopoulos, R. Rosati, M. Savic, C. Badica, M. Radovanovic (Eds.), Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS 2018, Novi Sad, Serbia, June 25-27, 2018, ACM, 2018, pp. 1:1–1:6. URL: https://doi.org/10.1145/3227609.3227689. doi:10.1145/3227609.3227689.

[2] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, S. Auer, Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge, in: M. Kejriwal, P. A. Szekely, R. Troncy (Eds.), Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019, ACM, 2019, pp. 243–246. URL: https://doi.org/10.1145/3360901.3364435. doi:10.1145/3360901.3364435.

[3] D. Li, V. Yadav, Z. Afzal, G. Tsatsaronis, Unsupervised dense retrieval for scientific articles, in: Y. Li, A. Lazaridou (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: EMNLP 2022 - Industry Track, Abu Dhabi, UAE, December 7 - 11, 2022, Association for Computational Linguistics, 2022, pp. 313–321. URL: https://doi.org/10.18653/v1/2022.emnlp-industry.32. doi:10.18653/V1/2022.EMNLP-INDUSTRY.32.

[4] L. Yang, H. Chen, Z. Li, X. Ding, X. Wu, Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling, IEEE Trans. Knowl. Data Eng. 36 (2024) 3091–3110. URL: https://doi.org/10.1109/TKDE.2024.3360454. doi:10.1109/TKDE.2024.3360454.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[6] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, CoRR abs/2312.10997 (2023). URL: https://doi.org/10.48550/arXiv.2312.10997. doi:10.48550/ARXIV.2312.10997.

[7] A. Kaplan, J. Keim, M. Schneider, R. Reussner, Replication package for HubLink: A novel question answering retrieval approach over knowledge graphs, 2025. URL: https://doi.org/10.5281/zenodo.17036243. doi:10.5281/zenodo.17036243.

[8] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T. Kuo, N. Guan, C. J. Xue, Retrieval-augmented generation for natural language processing: A survey, CoRR abs/2407.13193 (2024). URL: https://doi.org/10.48550/arXiv.2407.13193. doi:10.48550/ARXIV.2407.13193. arXiv:2407.13193.

[9] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, CoRR abs/2401.08281 (2024). URL: https://doi.org/10.48550/arXiv.2401.08281. doi:10.48550/ARXIV.2401.08281. arXiv:2401.08281.

[10] V. R. Basili, D. M. Weiss, A methodology for collecting valid software engineering data, IEEE Transactions on Software Engineering 10 (1984) 728–738. URL: https://doi.org/10.1109/TSE.1984.5010301. doi:10.1109/TSE.1984.5010301.

[11] V. R. B. G. Caldiera, H. D. Rombach, The goal question metric approach, Encyclopedia of software engineering (1994) 528–532.

[12] C.-Y. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, in: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (WAS 2004), Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.pdf.

[13] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[14] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating Text Generation with BERT (2020). URL: https://openreview.net/forum?id=SkeHuCVFDr.

[15] S. Es, J. James, L. Espinosa Anke, S. Schockaert, RAGAs: Automated evaluation of retrieval augmented generation, in: N. Aletras, O. De Clercq (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, St. Julians, Malta, 2024, pp. 150–158. doi:10.18653/v1/2024.eacl-demo.16.

[16] OpenAI, text-embedding-3-large, https://platform.openai.com/docs/models/text-embedding-3-large, 2024. Accessed: 2025-09-01.

[17] A. Alinejad, K. Kumar, A. Vahdat, Evaluating the retrieval component in llm-based question answering systems, 2024. URL: https://arxiv.org/abs/2406.06458.
[18] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, Z. Liu, Evaluation of retrieval-augmented generation: A survey, in: W. Zhu, H. Xiong, X. Cheng, L. Cui, Z. Dou, J. Dong, S. Pang, L. Wang, L. Kong, Z. Chen (Eds.), Big Data, Springer Nature Singapore, 2025, pp. 102–120.
[19] OpenAI, gpt-4o-mini model, https://platform.openai.com/docs/models/gpt-4o-mini, 2024. Accessed: 2025-09-01.
[20] D. Banerjee, S. Awale, R. Usbeck, C. Biemann, Dblp-quad: A question answering dataset over the dblp scholarly knowledge graph, 2023. URL: https://arxiv.org/abs/2303.13351.
[21] D. Q. Nguyen, D. Q. Nguyen, S. B. Pham, Ripple down rules for question answering, Semantic Web 8 (2017) 511–532. doi:10.3233/SW-150204.
[22] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, in: Guide to advanced empirical software engineering, Springer, 2008, pp. 285–311.
[23] J. T. Dillon, The classification of research questions, Review of Educational Research 54 (1984) 327–361. doi:10.3102/00346543054003327.
[24] S. K. Ratan, T. Anand, J. Ratan, Formulation of research question - stepwise approach, J. Indian Assoc. Pediatr. Surg. 24 (2019) 15–20.
[25] M. Dubey, D. Banerjee, A. Abdelkawi, J. Lehmann, Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, 2019, pp. 69–78.
[26] V. Bolotova, V. Blinov, F. Scholer, W. B. Croft, M. Sanderson, A non-factoid question-answering taxonomy, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 1196–1207. doi:10.1145/3477495.3531926.
[27] M. Y. Jaradeh, M. Stocker, S. Auer, Question answering on scholarly knowledge graphs, in: M. Hall, T. Merčun, T. Risse, F. Duchateau (Eds.), Digital Libraries for Open Knowledge, Springer International Publishing, 2020, pp. 19–32.
[28] M. Konersmann, A. Kaplan, T. Kühn, R. Heinrich, A. Koziolek, R. H. Reussner, J. Jürjens, M. al-Doori, N. Boltz, M. Ehl, D. Fuchß, K. Großer, S. Hahner, J. Keim, M. Lohr, T. Saglam, S. Schulz, J. Töberg, Evaluation methods and replicability of software architecture research objects, in: 19th IEEE International Conference on Software Architecture, ICSA 2022, Honolulu, HI, USA, March 12-15, 2022, IEEE, 2022, pp. 157–168. URL: https://doi.org/10.1109/ICSA53651.2022.00023. doi:10.1109/ICSA53651.2022.00023.
[29] M. Konersmann, A. Kaplan, T. Kühn, 2022. URL: https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice/-/wikis/Data-Extraction, [Last accessed on 2025-08-01].
[30] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering 36 (2024) 3580–3599. URL: https://doi.org/10.1109/TKDE.2024.3352100. doi:10.1109/TKDE.2024.3352100.
[31] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph retrieval-augmented generation: A survey, CoRR abs/2408.08921 (2024). URL: https://doi.org/10.48550/arXiv.2408.08921. doi:10.48550/ARXIV.2408.08921. arXiv:2408.08921.
[32] J. Baek, A. F. Aji, J. Lehmann, S. J. Hwang, Direct fact retrieval from knowledge graphs without entity linking, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 10038–10055. URL: https://aclanthology.org/2023.acl-long.558/. doi:10.18653/v1/2023.acl-long.558.
[33] Y. Sui, Y. He, N. Liu, X. He, K. Wang, B. Hooi, Fidelis: Faithful reasoning in large language models for knowledge graph question answering, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, Association for Computational Linguistics, 2025, pp. 8315–8330. URL:

https://aclanthology.org/2025.findings-acl.436/.

[34] Y. Wen, Z. Wang, J. Sun, MindMap: Knowledge graph prompting sparks graph of thoughts in large language models, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 10370–10388. URL: https://aclanthology.org/2024.acl-long.558/. doi:10.18653/v1/2024.acl-long.558.

[35] D. C. Montgomery, Design and analysis of experiments, John wiley & sons, 2017.

[36] OpenAI, gpt-4o model, platform.openai.com/docs/models/gpt-4o, 2024. Accessed: 2025-09-01.

[37] OpenAI, 03 mini model, platform.openai.com/docs/models/o3-mini, 2025. Accessed: 2025-09-01.

[38] Q. Team, Qwen2.5: A party of foundation models, 2024. URL: https://qwenlm.github.io/blog/qwen2.5/.

[39] A. Grattafiori, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[40] S. Lee, A. Shakir, D. Koenig, J. Lipp, Open source strikes bread - new fluffy embedding model, 2024. URL: https://www.mixedbread.ai/blog/mxbai-embed-large-v1.

[41] I. Granite Team, Granite 3.0 language models, 2024. URL: https://github.com/ibm-granite/granite-3.0-language-models/, accessed: 2025-09-01.

[42] L. Zhang, J. Zhang, X. Ke, H. Li, X. Huang, Z. Shao, S. Cao, X. Lv, A survey on complex factual question answering, AI Open 4 (2023) 1–12. URL: https://doi.org/10.1016/j.aiopen.2022.12.003. doi:10.1016/J.AIOPEN.2022.12.003.

[43] M. Y. Jaradeh, M. Stocker, S. Auer, Question answering on scholarly knowledge graphs, in: M. M. Hall, T. Mercun, T. Risse, F. Duchateau (Eds.), Digital Libraries for Open Knowledge - 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25-27, 2020, Proceedings, volume 12246 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 19–32. URL: https://doi.org/10.1007/978-3-030-54956-5_2. doi:10.1007/978-3-030-54956-5_2.

[44] D. Banerjee, S. Awale, R. Usbeck, C. Biemann, Dblp-quad: A question answering dataset over the DBLP scholarly knowledge graph, in: I. Frommholz, P. Mayr, G. Cabanac, S. Verberne, J. Brennan (Eds.), Proceedings of the 13th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 45th European Conference on Information Retrieval (ECIR 2023), Dublin, Ireland, April 2nd, 2023, volume 3617 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 37–51. URL: https://ceur-ws.org/Vol-3617/paper-05.pdf.

[45] M. Li, H. Yang, Z. Liu, M. M. Alam, H. Sack, G. A. Gesese, et al., Kgmistral: Towards boosting the performance of large language models for question answering with knowledge graph integration, in: Workshop on Deep Learning and Large Language Models for Knowledge Graphs, 2024. URL: https://www.fiz-karlsruhe.de/sites/default/files/FIZ/Dokumente/Forschung/ISE/Publications/Conferences-Workshops/7-KGMistral-Towards-Boosting-t.pdf.

[46] T. A. Taffa, R. Usbeck, Leveraging llms in scholarly knowledge graph question answering, in: QALD/SemREC@ ISWC, 2023. URL: https://ceur-ws.org/Vol-3592/paper5.pdf.

[47] J. Lehmann, A. Meloni, E. Motta, F. Osborne, D. R. Recupero, A. A. Salatino, S. Vahdati, Large language models for scientific question answering: An extensive analysis of the sciqa benchmark, in: A. Meroño-Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web - 21st International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26-30, 2024, Proceedings, Part I, volume 14664 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 199–217. URL: https://doi.org/10.1007/978-3-031-60626-7_11. doi:10.1007/978-3-031-60626-7_11.

[48] L. Jiang, X. Yan, R. Usbeck, A structure and content prompt-based method for knowledge graph question answering over scholarly data, in: D. Banerjee, R. Usbeck, N. Mihindukulasooriya, G. Singh, R. Mutharaju, P. Kapanipathi (Eds.), Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023, volume 3592 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3592/paper3.pdf.

[49] Y. Gu, V. Pahuja, G. Cheng, Y. Su, Knowledge base question answering: A semantic parsing

perspective, in: S. Riedel, E. Choi, A. Vlachos, J. Thorne, M. Rei, F. Petroni (Eds.), 4th Conference on Automated Knowledge Base Construction, AKBC 2022, London, UK, November 3-5, 2022, 2022. URL: https://akbc.ws/2022/papers/23_knowledge_base_question_answer.

[50] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. M. Ni, H. Shum, J. Guo, Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net, 2024. URL: https://openreview.net/forum?id=nnVO1PvbTv.

[51] C. Feng, X. Zhang, Z. Fei, Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs, CoRR abs/2309.03118 (2023). URL: https://doi.org/10.48550/arXiv.2309.03118. doi:10.48550/ARXIV.2309.03118. arXiv:2309.03118.

[52] L. Sun, Z. Tao, Y. Li, H. Arakawa, ODA: observation-driven agent for integrating llms and knowledge graphs, in: L. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024, Association for Computational Linguistics, 2024, pp. 7417–7431. URL: https://doi.org/10.18653/v1/2024.findings-acl.442. doi:10.18653/V1/2024.FINDINGS-ACL.442.

[53] B. Jin, C. Xie, J. Zhang, K. K. Roy, Y. Zhang, Z. Li, R. Li, X. Tang, S. Wang, Y. Meng, J. Han, Graph chain-of-thought: Augmenting large language models by reasoning on graphs, in: L. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024, Association for Computational Linguistics, 2024, pp. 163–184. URL: https://doi.org/10.18653/v1/2024.findings-acl.11. doi:10.18653/V1/2024.FINDINGS-ACL.11.

[54] J. Jiang, K. Zhou, Z. Dong, K. Ye, X. Zhao, J. Wen, Structgpt: A general framework for large language model to reason over structured data, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 9237–9251. URL: https://doi.org/10.18653/v1/2023.emnlp-main.574. doi:10.18653/V1/2023.EMNLP-MAIN.574.

[55] S. Ma, C. Xu, X. Jiang, M. Li, H. Qu, C. Yang, J. Mao, J. Guo, Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation, in: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025, OpenReview.net, 2025. URL: https://openreview.net/forum?id=oFBu7qaZpS.

[56] Y. Xu, S. He, J. Chen, Z. Wang, Y. Song, H. Tong, K. Liu, J. Zhao, Generate-on-graph: Treat LLM as both agent and KG in incomplete knowledge graph question answering, CoRR abs/2404.14741 (2024). URL: https://doi.org/10.48550/arXiv.2404.14741. doi:10.48550/ARXIV.2404.14741. arXiv:2404.14741.

[57] J. Kim, Y. Kwon, Y. Jo, E. Choi, KG-GPT: A general framework for reasoning on knowledge graphs using large language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 9410–9421. URL: https://doi.org/10.18653/v1/2023.findings-emnlp.631. doi:10.18653/V1/2023.FINDINGS-EMNLP.631.

[58] B. Jiang, Y. Wang, Y. Luo, D. He, P. Cheng, L. Gao, Reasoning on efficient knowledge paths: Knowledge graph guides large language model for domain question answering, in: V. S. Sheng, C. Hicks, C. Ling, V. Raghavan, X. Wu (Eds.), IEEE International Conference on Knowledge Graph, ICKG 2023, Shanghai, China, December 1-2, 2023, IEEE, 2024, pp. 142–149. URL: https://doi.org/10.1109/ICKG63256.2024.00026. doi:10.1109/ICKG63256.2024.00026.

[59] Z. Zhang, X. Liu, Y. Zhang, Q. Su, X. Sun, B. He, Pretrain-kge: Learning knowledge representation from pretrained language models, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of *Findings of ACL*, Association for Computational Linguistics, 2020, pp. 259–266. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.25. doi:10.18653/V1/2020.FINDINGS-EMNLP.25.

[60] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, Trans. Assoc. Comput. Linguistics 9 (2021) 176–194. URL: https://doi.org/10.1162/tacl_a_00360. doi:10.1162/TACL_A_00360.

[61] M. Nayyeri, Z. Wang, M. M. Akter, M. M. Alam, M. R. A. H. Rony, J. Lehmann, S. Staab, Integrating knowledge graph embedding and pretrained language models in hypercomplex spaces, CoRR abs/2208.02743 (2022). URL: https://doi.org/10.48550/arXiv.2208.02743. doi:10.48550/ARXIV.2208.02743. arXiv:2208.02743.