RAG and Ontologies for Information Retrieval: A Literature Review

Bart Gajderowicz[1,2,*], Aviral Bhardwaj[3] and Mark Fox[1]

[1]Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada
[2]School of Computing, Utah State University, Logan, Utah, USA
[3]Computer Science, University of Toronto, Toronto, Ontario, Canada

Abstract

As large language models (LLMs) increasingly serve as general-purpose information tools, the integration of structured semantic resources, particularly ontologies and knowledge graphs, into retrieval-augmented generation (RAG) systems has become a key area of exploration. This literature review examines the role of ontologies in enhancing information retrieval tasks within RAG systems, particularly when combined with LLMs. We examine how ontological structures enhance retrieval quality, support validation and verification. Three primary research questions guide the review. First, we review the targeted application domains of ontology-driven retrieval in practice. Second, we analyze how ontologies are utilized in information retrieval with RAG, focusing on the tasks that are performed and methods used. Third, we evaluate the data (input) that is required for information retrieval using RAG, the data produced (output), evaluate the results, and outline limitations.

Keywords

retrieval-augmented generation, information retrieval, knowledge graph, ontology, large language models

1. Introduction

As large language models (LLMs) continue to advance as general-purpose tools for information access, the question of how to integrate structured semantic resources, particularly ontologies and knowledge graphs (KG), into these systems has become increasingly salient. Retrieval-Augmented Generation (RAG) offers a hybrid architecture that grounds language model outputs in external knowledge sources, aiming to enhance factuality, relevance, and interpretability. Within this framework, ontologies and KGs play a potentially critical role in improving information retrieval, providing structured representations that encode domain-specific semantics, support query disambiguation, and enable more reliable validation of generated responses. For the purposes of this review, we consider ontologies as the schema for concepts and relationships, while a KG is a graph database that uses an ontology as its schema. Different articles use both terms, and unless specified, we use terminology used in the article referenced.

This literature review investigates how ontologies and KGs are used to enhance retrieval tasks in RAG systems, especially when paired with LLMs. It focuses on the structural and semantic contributions of ontologies to retrieval quality, and the extent to which they support processes such as validation, verification, and reasoning. The review is guided by three primary research questions:

RQ1 : In what applications are ontologies used for RAG, in the presented methods.

RQ2 : How are ontologies used with RAG in information retrieval?

Convergence of Large-Language Models and Ontologies (ONTOLLM), Proceedings of the Joint Ontology Workshops (JOWO) - Episode XI: The Sicilian Summer under the Etna, co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 8-9, 2025, Catania, Italy

*Corresponding author.

† **Declaration on Generative AI**: The author used Grammarly system for spell checking and ChatGPT to reduce text of some paragraphs, reviewed/edited the content, and take full responsibility for the publication's content.

bart.gajderowicz@utoronto.ca (B. Gajderowicz); aviral.bhardwaj@mail.utoronto.ca (A. Bhardwaj); msf@eil.utoronto.ca (M. Fox)
https://bartg.org/ (B. Gajderowicz); http://www.eil.utoronto.ca (M. Fox)
0000-0001-6201-8781 (B. Gajderowicz); 0000-0001-7444-6310 (M. Fox)

RQ3 : What methods combine ontologies and LLMs for information retrieval?

- **RQ3.1** : What data about the information must be provided (as input) to retrieval, validation, and verification systems?
- **RQ3.2** : What data is produced (as output) as retrieved information?
- **RQ3.3** : What are the results and limitations of information retrieval using ontologies?

Together, these questions aim to clarify the evolving relationship between structured knowledge and generative language technologies, as utilized by retrieval-augmented generation techniques, providing a grounded assessment of the potential and limitations of ontology-driven retrieval in the age of LLMs.

1.1. Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is an advanced hybrid architecture in natural language processing (NLP) that integrates external retrieval mechanisms with generative language models to enhance the accuracy and factual grounding of generated outputs. RAG systems typically involve two core components: a retrieval module that identifies and fetches relevant information from external knowledge sources, such as documents or databases, accessed as dense vector representations or through traditional retrieval methods, and a generation module, often transformer-based, which synthesizes the retrieved information into coherent and contextually relevant responses [1, 2]. By dynamically grounding outputs in up-to-date external information, RAG effectively mitigates common limitations of traditional LLMs, such as hallucinations and outdated knowledge [1].

The integration of retrieval and generation components enables RAG models to excel in knowledge-intensive tasks such as open domain question answering (Q/A), summarization, conversational agents, and personalized recommendations, providing significantly improved factual accuracy compared to models purely generative or retrieval-only [3, 4]. However, RAG systems face challenges related to the quality of retrieved documents, the alignment between retrieved content and generated output, the computational overhead associated with real-time retrieval processes, and potential biases introduced through retrieved documents [5, 6]. Ongoing research actively addresses these challenges, aiming to balance retrieval accuracy with the quality of generated results.

1.2. Word Embeddings and Semantic Information

Information stored as an ontology or KG is meant to address some of the challenges with LLMs by providing explicit knowledge in the form of "semantic information," rather than implicit meaning and token patterns in natural language text. By "semantic information" we mean any information used for semantic similarity search [7] as part of the embedding [8], including contextual information [9], semantic patterns unique to a domain [10], as well as graph-based and model-theoretic strategies [8]. A key performance and speed boost LLMs have in recent years comes from the type of embedding method used to embed text into a vector representation, as well as scaling and parallelizing computational resources. Table 1 displays the most common embedding types and their characteristics, highlighting the role semantic information plays in each embedding method [11, 12].

Frequency-based embeddings, such as one-hot-encoding for efficient frequency of categories, TF-IDF (term-frequency/inverse-document-frequency), and LAS (Latent Semantic Analysis) rely on the frequency of words to make associations between tokens across documents [13]. These methods capture a low-degree of semantics. Prediction-based embedding type, such as Word2Vec and FastText, capture relationships between co-occurrence of tokens in the local context (proximity to other tokens), providing a medium level of semantics from word usage in similar phrases [14, 12]. While the encoding is fast due to smaller context windows, these methods create static embeddings, meaning that embedding weights between words do not change in different contexts during the testing stage [14]. Hybrid methods, such as GloVe (Global Vectors), extend the context window to find global co-occurrences across the entire training corpus. Context-based methods, such as ELMo (sequence memory with Long-Short-Term-Memory method) and GPT-based (Generative Pre-trained Transformer, Left-to-right transformers) form

Table 1
Word Embeddings

Type	Embedding Method	Semantics Information	Context Handling	Static / Dynamic
Frequency-Based	One-Hot Encoding (of categories)	Very Low	None	Static
	TF-IDF	Low	None	Static
	Latent Semantic Analysis	Low–Medium	None (latent structure)	Static
Prediction-Based	Word2Vec	Medium	Local context (window)	Static
	FastText	Medium	Local context + subword	Static
Hybrid	GloVe	Medium–High	Global co-occurrence	Static
Context-Based	ELMo	High	Bi-directional (LSTM)	Dynamic
	GPT (GPT-2, GPT-3, etc)	High	Left-to-right (Transformer)	Dynamic
	BERT (RoBERTa, DistilBERT)	Very High	Bi-directional (Transformer)	Dynamic

a complex weight network of word embeddings [12, 14]. By training the embedding of each token in different contexts, these methods are dynamic, meaning, the embedding weights change based on the surrounding words during the testing phase [14, 11]. BERT-based methods (Bidirectional Encoder Representations from Transformers), such as RoBERTa, SBERT, and DistilBERT, have bi-directional transformers, and provide a higher quality of semantic relationships between words [9, 15]. As can be seen, the better performing models, GPT-based and BERT-based, have high and very high quality of semantic relationships between tokens [15]. However, these are latent semantics inferred from the text itself. By incorporating ontologies and KGs, the quality of semantic relationships can be increased to ensure that explicit and intended semantics are used on specific tasks and domains.

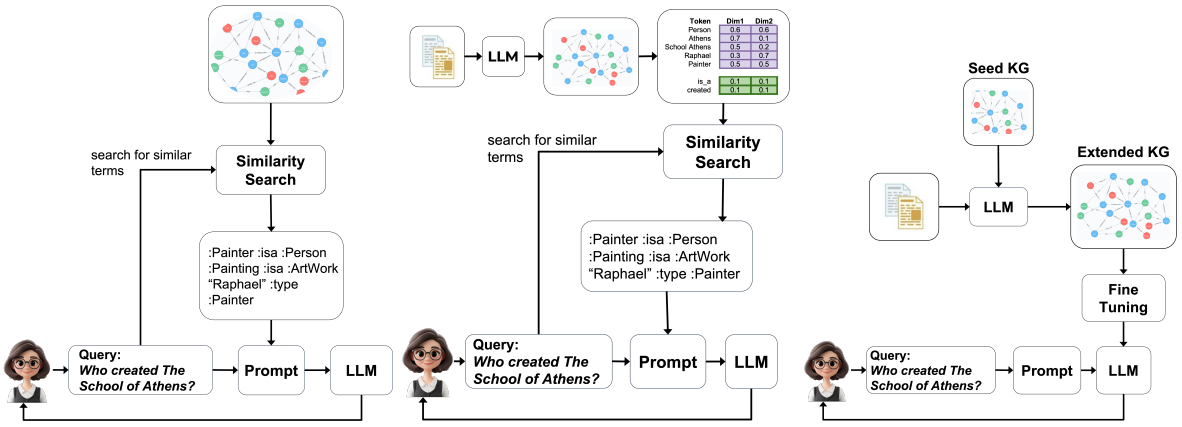
Some techniques rely on graph embeddings of ontologies or KGs as well as word embeddings. In such systems, the distinction between ontologies/KGs and "graphs" is that graph embeddings retain the structure of the relationships between classes and instances, but lose the semantics defined by the ontology, retained only in RDF-based graphs. Embeddings of such graphs rely on various graph traversal algorithm (e.g., random walks, Breadth-first-search) used as part of the embedding learning process. However, once a graph is embedded, the resulting vectors do not support traditional traversal; they enable similarity and clustering tasks in the vector space.

1.3. RAG+KG Techniques

RAG with ontologies or KGs is based on three main paradigms, each one utilizing specific knowledge in the graphs and improving LLM-generated answers. Combing KGs with LLMs (Figure 1a) involves identifying key terms in the user's query and searching a KGs for similar terms. The similarity search method can compare terms in the query with classes, instances, and predicates in the KG. Once similar terms are found, these are added to the prompt to provide context before sending them to the LLM for processing.

The GraphRAG approach (Figure 1b) involves extracting a KG out of raw text, and organizing the embeddings into a community hierarchy. The KG will self-organize terms that are closely related and through semantic meaning and co-occurrences. The communities form context for terms that are matched to terms in the user's query. These communities represent a type of summary for the content, capturing key terms in a compact and localized manner. These structures are leveraged when performing RAG-based tasks such as similarity search and subgraph extraction.

The third technique (Figure 1c) extends the GraphRAG technique by fine tuning both the KG and then the LLM, forming a closer bond between the two sources of information. First, the KG is "fine tuned" by extending the seed ontology with link predictions created by the LLM, stored as the extended KG. Second, the LLM is fine-tuned with new facts inferred from the extended KG. For example, an



(a) LLMs + KG + RAG. (b) GraphRAG. (c) Fine-Tuning RAG and LLMs.

Figure 1: RAG Methodologies

LLM can be used to embed a set of documents. These embeddings are combined with a "seed" KG that contains an incomplete graph with some classes and instances in the domain of interest. The result is an extended KG with two characteristics. First, the terms used for class names and predicate labels are the same or similar to those in the document corpus. Second, it performs graph completion by creating new classes and instances, and edge completion by creating new relationships between existing classes, instances, or a combination of both. A third benefit of this approach is that it creates hypergraphs within the KG [16, 17], significantly increasing the quality of information retrieval in generated content.

2. Review Methodology

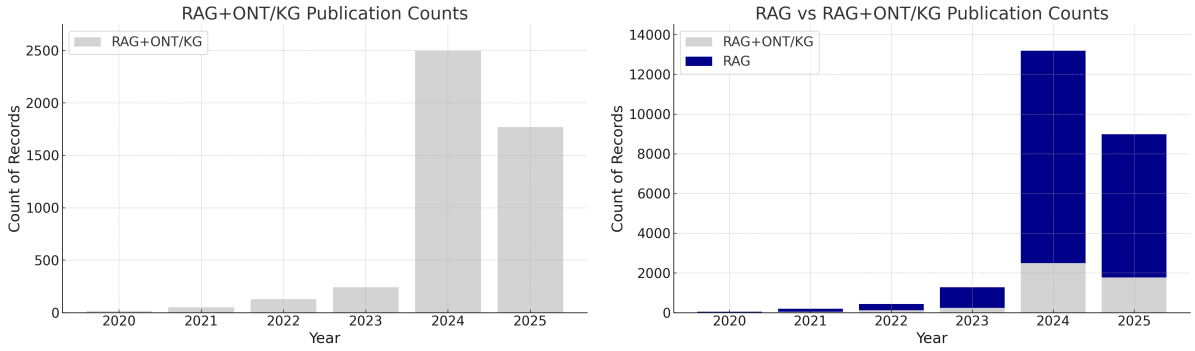
In this work, we set out to review how ontologies and KGs are utilized to perform information retrieval using LLMs. To answer the RQs, we used the Google Scholar database. Considering the specific RQs and the novelty of RAG systems and LLMs, we did not limit the search to any domain. The resulting query searches scientific literature for: (("ontology" OR "knowledge graph") AND "Retrieval-Augmented Generation") focusing on publication years $>$2020 and $\leq$2025, and limiting the search to the "Keywords" search field which looked at "Title," "Abstract," and snippets from the "body" of the publication of each record. The total results were 4,641 publications.

The database search was conducted in April, 2025. Articles selection process and criteria focused on a combination of most cited, included quantitative evaluation section, and a variety of methodologies that utilize ontologies and KGs for RAG tasks to improve information retrieval from LLMs. Some included papers had a low citation count, as they were published between late 2024 and early 2025, but introduced a novel methodology with a comprehensive evaluation or discussion section. Hence, a strict citation cutoff for inclusion was not used, rather, top 30 cited papers were reviewed under these criteria. The resulting 20 publications capture a variety of methods that combine ontologies and KGs with LLMs for the improvement of information retrieval, with comprehensive validation and verification sections.

3. Results

3.1. Temporal distribution of articles by year

The temporal distribution of the works included in this review (Figure 2a) indicates that only 16 contributions addressing the topics pertaining to the RQs in 2020, with a significant increase in 2024, with 2,500 records. This is expected as the term "retrieval-augmented generation" is relatively new, although the technique has been used in a number of publications. Not surprisingly, most relevant articles are found in the years 2024 and 2025 with the popularity of LLMs. In previous years, the RAG technique was applied to traditional NLP methods for knowledge-intensive tasks [1, 18]



(a) Publication counts including RAG and Ontologies or KGs.

(b) Publication counts comparing RAG vs RAG and Ontologies or KGs.

Figure 2: Publication Counts

To show the relevance of ontologies and KGs, Figure 2b illustrates the same data, but as a portion of all RAG-related publications in the same time frame. We see that ontologies and KGs make up 16% of publications related to RAG methodologies.

3.2. Applications of Ontology-driven RAG

To answer research question RQ1, we evaluate the applications of RAG across different domains.

3.2.1. Scientific-engineering workflow documentation and design optimization

Ontology-guided retrieval has been adopted to capture complex research workflows and engineering processes. In biodiversity-oriented deep-learning pipelines, an ontology aligned with DLProv and PROV-O classes records datasets, model stages, and hardware so that RAG queries can automatically document experiments and trace provenance for reproducibility [19]. Additive-manufacturing analytics represent process parameters and their inter-dependencies as a graph whose schema comprises domain classes as nodes and typed edges; subgraph retrieval enriches prompts for context-dependent design decisions [20]. Materials-science research employs a KG whose nodes denote materials, mechanisms, or behaviors and whose edges capture causal or compositional relations, supporting automated materials design, protein-mechanics modelling, and force-field development [21].

3.2.2. Biomedical annotation, diagnostics, and ontology curation

Clinical and life-science applications rely on ontologies to normalise terminology and surface evidence. The human phenotype ontology (HPO) includes phenotype annotation that link sentence embeddings to metadata within a graph, enabling rapid retrieval of candidate terms for genomic diagnostics and genotype–phenotype studies [22]. DRAGON-AI helps curators extend biomedical and environmental ontologies by retrieving similar term embeddings from a ChromaDB vector store; the resulting JSON-schema graph records label, definition, and relationships fields for each candidate term[23]. A dental-materials search portal fuses domain ontologies such as the Dental Restorative Material Ontology (DrMo) and the Oral Health and Disease Ontology (OHD) [24] with metadata from Dublin Core and BIBO [25], as well as Prov [26]. The resulting ontology exposes relations like *hasAuthor*, *isAbout*, and *isAboutProduct*, supporting evidence-based clinical decisions [27]. General KG-enhanced frameworks also target drug discovery by retrieving entity descriptions and hierarchical clusters that mix structural links with LLM semantics [17].

3.2.3. Ontology-Driven Applications in Education and Enterprise Systems

Ontology-driven information retrieval has demonstrated promising applications in both academic and enterprise environments. In educational contexts, KGs composed of entities such as *Student*, *Course*, *Faculty*, *Policy*, and *Intent*, and relations like *enrolledIn*, *requires*, or *pertainsTo*, have been employed to support campus-wide automation, academic Q/A, and intent classification. These graphs feed Cypher queries to power back-end reasoning for academic services and student support systems [28]. Similarly, in enterprise settings, ontologies represented in the Web Ontology Language (OWL) [29] mapped from relational schemas using R2RML enable Ontology-Based Query Checks. Here, the SPARQL Protocol and RDF Query Language (SPARQL) [30] rule violations are automatically corrected and revalidated by an LLM, facilitating reliable Q/A over SQL (Structured Query Language) databases. This mechanism supports structured query interpretation and ensures data integrity across business applications [31].

3.2.4. Ontology-Guided Summarization and Social Research

Beyond structured environments, ontology-enhanced methods have also been applied to summarization tasks and social-policy research. GraphRAG, for example, facilitates global, query-focused summarization by constructing KGs from unstructured sources such as podcasts and public-health news. These graphs, structured with entity nodes, relation edges, and claim properties, are partitioned via community detection, and then summarized hierarchically to provide concise, query-relevant responses [32]. In the social sciences, structured survey codebooks are parsed into entity-relationship graphs within systems like PostgreSQL, where tables such as Substance, Incident, and Person, along with foreign-key constraints, preserve contextual relationships. These representations underpin ontology-enhanced retrieval for tasks such as substance-use analytics and public-policy investigations, enabling data-driven insights grounded in structured semantics [33].

3.2.5. Structured-workflow domains and professional knowledge work

OG-RAG demonstrates that agriculture, healthcare, legal practice, and journalism benefit from ontologies or semantically rich graphs that encode entities and hierarchical abstractions of domain workflows, allowing retrieval of concise fact clusters that ground LLM reasoning [16]. Add-on surveys highlight similar applications in recommendation, commonsense reasoning, and medical diagnosis, using KG triples, embeddings, and reasoning paths aligned with text [20].

3.2.6. Multi-hop question answering and reasoning over open KGs

A group of methods, including Neural State Machine, Think-on-Graph, GRAG, KG-FIT, and KG-planner, targets open-domain Q/A, slot filling, fact-checking, and complex logical queries. The underlying graphs typically follow the Freebase or Wikidata style of entity–relation–entity triples, sometimes augmented with logical templates for intersections, unions, or comparisons [34, 35, 36, 37, 38, 39]. Schema extensions may include textual descriptions, hierarchical clusters, or sentence-level edge embeddings to capture LLM-generated semantic relations [17, 40].

3.3. Uses of Ontology-driven RAG

To answer research question RQ2, we evaluate the ways ontology-driven RAG is used.

3.3.1. Knowledge-graph and ontology construction for enhanced retrieval

RAG can be used to build the very knowledge assets that later drive retrieval. A semi-automated pipeline synthesizes biodiversity ontologies and KGs by seeding construction with LLM-based information extraction [19]. In biomedical curation, DRAGON-AI employs LLM-driven RAG to draft textual and logical definitions that complete terms across multiple life-science ontologies[23]. StructuGraphRAG

converts structured survey codebooks into task-specific KGs that underpin substance-use and mental-health research [33]. A recent survey groups these efforts into three archetypes, KG-enhanced LLMs, LLM-augmented KGs, and fully synergistic LLM + KG systems, highlighting complementary routes to richer representation and retrieval [41].

3.3.2. Ontology-driven semantic search and question answering

Embedding ontological structure directly into RAG pipelines boosts semantic search and Q/A. An enterprise-SQL assistant fixes faulty SPARQL by reasoning over an ontology-backed KG, raising answer accuracy [31]. A cross-institutional campus graph supports a KG-augmented Q/A system that provides detailed responses to students and staff [28]. For Indian dental clinicians, a semantic search tool fuses an LLM with a material-product ontology to deliver evidence-based guidance on restoratives and devices [27]. OG-RAG shows that domain-specific ontologies can raise retrieval precision across healthcare, agriculture, law, journalism, and consulting workflows [16].

3.3.3. Domain-specific RAG for scientific and technical information

Ontology-aware RAG is being tailored to specialized scientific corpora. In additive-manufacturing analytics, an ontology-based framework enriches prompts so that generated output aligns with process parameters and material contexts [20]. MechGPT, fine-tuned on materials-science literature, combines RAG with a KG to support hypothesis generation, mechanistic discovery, and code-assisted force-field development [21]. Automated phenotype extraction mines Human Phenotype Ontology terms from clinical narratives, scaling genomic diagnostics that hinge on precise term retrieval [22]. GraphRAG builds a hierarchical KG over large text corpora to enable query-focused summarization and global sense-making across disparate sources [32].

3.3.4. Improving knowledge-graph embeddings for retrieval and link prediction

Injecting global semantic priors from LLMs into structure-based embeddings (KG-FIT) markedly increases link-prediction accuracy on benchmarks such as FB15K-237, YAGO3-10, and PrimeKG, directly benefiting retrieval over incomplete graphs [17]. A three-stage zero-shot pipeline of alignment, reasoning, and reranking further strengthens entity prediction in sparse and dense graphs by integrating frozen LLMs [40].

3.3.5. Graph-guided reasoning and multi-hop question answering

Ontologies and KGs can serve as external reasoning workspaces for multi-step queries. Think-on-Graph iteratively explores KG paths with beam search, injecting retrieved triples into the LLM context to raise multi-hop accuracy and provide transparent rationales [34, 35]. GRAG extends this idea to text-attributed graphs for commonsense and multi-hop Q/A [37]. Intermediate supervision in a teacher–student framework improves KG-based Q/A reasoning [36], and ontology-based path construction sharpens reasoning efficiency [38]. Training LLMs on planning traces distilled from KGs yields better decomposition of complex questions into executable plans [39].

3.4. Methods for Information Retrieval with RAG

To answer the research question RQ3, we identify the methods used to retrieve information from various sources and formats that incorporate structured knowledge in the form of ontologies or KGs. Recent advancements in RAG methodologies that utilize ontologies and KGs have produced diverse technologies designed to enhance information retrieval, structured representation, and reasoning. These technologies focus on structured knowledge modelling, hybrid query understanding, and optimized retrieval strategies that enable LLMs to effectively access and reason over domain-specific knowledge.

3.4.1. Ontology and Knowledge Graph Construction

Ontology-to-graph transformation technologies extract classes and individuals from ontologies and represent them as nodes connected by relational edges [20]. Many RDF-based graph representations (e.g., OWL or RDFS) retain formal semantics, but many of those in use do not (e.g. Neo4j). Regardless of the graph representation, the resulting embeddings do not retain the semantics and do not support reasoning, relying instead on any latent semantics for similarity or relatedness calculations [7, 42]. Semantic loss is not intrinsic to graph conversion from ontology to KG but depends on the representation layer where logical constraints and axioms are lost during the vectorization process. Embeddings retain topological relationships but abstract away some axiom-level details. As discussed below and in section 3.4.5, fine-tuning offers better reasoning-like capabilities [38].

Embeddings of such graphs rely on graph traversal algorithms rather than reasoning. Entity and relationship extraction (from document text chunks using LLMs) enables the generation of RDF triple-like graphs, with community detection algorithms applied to create hierarchical partitions and summaries [32]. Graph construction also benefits from document parsing tools that analyze structural elements (e.g., headings, figures, tables) and their hierarchical relationships to create semantically and structurally weighted graphs [33]. LLMs can convert unstructured text into graph representations that reflect contextual knowledge through subgraph traversal, often up to two hops deep [21].

Semi-automated pipelines for KG construction integrate LLMs such as Mixtral 8x7B across multiple stages. These include data curation, generation of competency questions (CQs) with ChatGPT-3.5, ontology creation using extracted relationships and concepts, and CQ answering via RAG with domain-specific literature. Evaluation is conducted through LLM-based scoring against human-labeled ground truth [19]. In education-focused implementations, sentence embeddings generated via PhoBERT are clustered using UMAP and HDBSCAN, followed by automatic cluster labeling and relation mapping through Sentence-BERT and TF-IDF re-ranking. Resulting entities and relations are structured as a KG, enabling Cypher-based subgraph retrieval for LLM-based Q/A [28].

Embedding-based indexing and information retrieval supports ontology auto-completion by relying on models like OpenAI's text-embedding-ada-002 and vector databases like ChromaDB, with LLMs completing partially filled terms via structured prompts [23]. Knowledge alignment tools use LLM-generated relations to enrich KGs, applying closed, open, and semi-closed domain strategies combined with semantic similarity scoring for refinement [40]. Fine-tuned LLMs such as MechGPT are also used for domain-specific KG generation in materials science, where they assist in contextualizing knowledge structures into traversable graphs. These systems may incorporate multi-agent frameworks with specialized components for retrieval, planning, review, and simulation [21].

3.4.2. Prompt Engineering in RAG

To answer research question RQ3 further, we identify prompt engineering methods in which RAG and ontologies/KGs are combined by generating prompts. Prompt engineering in ontology/KG-based RAG systems is not only about wording but also about embedding structured, validated, and semantically aligned knowledge into prompts, enabling LLMs to generate accurate, domain-specific, and explainable outputs. Combining structured domain knowledge into prompt generation enhances the contextual accuracy and factual grounding of LLMs [20, 41]. In these systems, ontologies are converted into highly connected graphs enabling efficient retrieval of relevant subgraphs containing entities and their relationships [20]. Retrieved subgraphs, representing domain-specific concepts such as "Structure Optimization" or "Build Plate Side", are embedded directly into prompts to guide generative reasoning. However, challenges remain in scaling prompts for large KGs and minimizing manual tuning [41].

Prompt construction processes in these systems typically involve knowledge acquisition, retrieval, and structured prompt assembly, with multiple variations in how KGs are built and used [40, 32, 17, 23]. Some approaches retrieve entities and relations from existing ontologies or domain KGs, aligning LLM outputs to predefined schemas through closed, semi-closed, or open-domain strategies [40]. Others dynamically construct KGs from raw text by prompting LLMs to extract entities, relationships, and

claims, with few-shot exemplars used to maintain domain alignment [32]. Others prompt LLMs directly to generate KG entity descriptions, embeddings, and refined hierarchical relationships [17, 23].

Ontology-derived prompts are validated through reasoning trace diagnostics and fact-checking frameworks such as LAMA and LLM-facteval [41]. Semantic clustering and hierarchy quality are assessed by prompting LLMs to evaluate and refine entity groupings [17]. Post-processing techniques parse generated outputs, removing invalid or unsupported links [23].

3.4.3. Retrieval and Graph-Augmented Query Processing

Graph retrievers can identify and extract relevant subgraphs from KGs using vector similarity or structural coherence measures [20, 39], while Maximal Marginal Relevance based selection ensures diversity and relevance in retrieval outputs [23]. Hierarchical retrieval mechanisms over hypergraphs, where hyperedges cluster related factual units, allow for multi-level reasoning over structured content [16]. Semantic path extraction and pruning methods use LLM evaluations to filter meaningful graph paths before answer generation [18]. Search-based retrieval strategies like beam search enable breadth- and depth-limited graph exploration, where LLM dynamically assesses path relevance [35]. Planning modules can also guide KG traversal by generating and optimizing relation paths, which are then used for constrained retrieval [34].

RAG systems benefit from hybrid representations that combine structural and semantic signals. For example, dual-view prompting frameworks use both hard and soft graph encoding to inform LLMs. Subgraphs are retrieved by identifying k-hop ego-graphs, weighted for relevance using Multilayer Perceptrons, and embedded through mean pooling of their node and edge attributes [37]. The "text view" reorders these subgraphs using breadth-first traversal into hierarchical prompts, while the "graph view" encodes them through GNNs aligned to LLM embeddings. These views are concatenated into a joint prompt for graph-aware response generation [37]. In clinical RAG systems, synthetic context sentences for Human Phenotype Ontology (HPO) terms are generated by ChatGPT-3.5, embedded using models such as MPNET and MultiQA-MiniLM, and stored in ChromaDB for similarity-based retrieval. Fusion models combine these LLM-based approaches with traditional NLP pipelines like PhenoTagger using majority voting to improve accuracy [22].

Prompt engineering and retrieval hyperparameters play a crucial role in optimizing performance. For instance, Mixtral 8x7B configurations may use a low temperature (10^5) and large token limits (25,000) to generate stable outputs, with document chunks sized at 2,500 tokens and 100-token overlaps to preserve continuity. Query templates like "Considering <CONTEXT>, answer <QUESTION>" structure the interaction between retrieved knowledge and LLMs [19, 21].

3.4.4. Query Interpretation

Entity extraction and query interpretation technologies leverage prompt-based approaches to identify domain-specific concepts within user queries [20]. Techniques like multi-hop reasoning and search grounded in extracted terms match query terms to ontology labels and instance data. This requires three main steps: 1) extract terms found in user questions and queries, 2) connect terms to instances, classes, or instance data in KG, and 3) form direct connections between instances or classes using properties in the KG. This provided a more targeted knowledge path retrieval within graphs, often implemented through recursive tree structures such as tree-of-thought and chain-of-thought [38]. Query processing technologies can also account for content and structural requirements (e.g., recognizing needs for tabular or sequential information), ensuring semantic and contextual relevance in reasoning tasks [33].

3.4.5. Reasoning Over Graphs

Reasoning technologies operate over retrieved content to ensure factual consistency and semantic coherence in generated responses. Guided answer mining pipelines use tree-based structures and semantic pruning to select and process graph paths, which are then passed into LLMs for final response generation [38]. Neural State Machine architectures model reasoning as transitions between state

representations of entity distributions, supervised through bidirectional reasoning and guided by teacher-student networks [36]. Combined planning-retrieval-reasoning frameworks maximize retrieval fidelity and answer quality by jointly optimizing planning and reasoning objectives [34]. LLM-guided generation approaches use retrieved knowledge as contextual grounding for producing accurate, context-aware outputs [39, 40].

3.5. Data

To answer the research questions RQ3.1 and RQ3.2, we identify the data that must be provided and the outputs that were produced.

Input: Regarding inputs, most systems rely on structured representations such as ontologies and KGs that encode domain-specific relationships and entities. For example, KGs were utilized to structure contextual knowledge for answering natural language queries [39, 18, 34], while domain-specific ontologies supported both deductive reasoning and ontology completion [16, 23]. Other systems enriched incomplete graphs using LLM-inferred relations [40] or used structured document metadata, such as tables and figures, for context-aware retrieval [33]. Common to all studies was the use of natural language queries, often submitted by users seeking domain-specific information [20, 36, 35], and hybrid systems that combined textual, graph-based, and model-driven inputs [32, 35].

These hybrid systems also included diverse implementations of RAG paired with ontologies and KGs. Input strategies varied, encompassing domain-specific corpora, biomedical datasets, structured ontologies, and learned embeddings. For example, domain literature was used to generate ontologies and KGs for conceptual query alignment [19], while a large-scale human phenotype ontology dataset was augmented with synthetic corpora created via ChatGPT [43] to support ontology-driven term extraction [22]. Vietnamese educational data was processed using a combination of local NLP tools and language models to uncover open-ended intents and construct KGs tailored to low-resource domains [28]. Other systems employed OWL ontologies for semantic validation [31], graph representations enhanced through Graph Attention Networks [37], and embedding pipelines for knowledge transfer without LLM fine-tuning [17]. Some frameworks also processed multi-modal inputs via multi-agent architectures integrated with ontological graph representations to support scientific discovery [21].

Outputs: The outputs from these systems were similarly varied yet consistently designed to provide contextually accurate and domain-aligned results. Systems generated direct answers accompanied by interpretable reasoning paths [36, 35, 38] and enriched knowledge artifacts such as completed or expanded ontologies in structured formats like JSON [40, 23]. In some cases, visualized subgraphs and query-specific summaries were produced to enhance user interpretability [32, 39, 33], while others focused on manufacturing documentation generation and contextual explanations [20]. Additional outputs included multi-source ontologies such as DLProv, structured SPARQL validators, Cypher-query-based Q/A systems, hybrid graph-based retrievers, and graph-enhanced embeddings [19, 31, 28, 17, 37]. Fused models combined dense retrievers with ontology-aware embeddings to improve term mapping and semantic precision [22], and multi-agent LLM systems returned deeper, contextually grounded responses in domains like biomedical and chemical design [21].

Results: The results consistently demonstrated improved performance across a range of evaluation metrics. Accuracy and multi-hop reasoning capabilities were significantly enhanced, with notable improvements in standard KG+Q/A benchmarks such as F1 and Hit@1 scores [38] and robust handling of complex multi-hop queries [16, 36]. Edge [32] and Zhu [33] report better query relevance and integration performance compared to traditional RAG systems. Methods that enriched KGs by generating new relations between edges using LLMs outperformed conventional baselines on sparse datasets and abstract relation tasks [40]. Frameworks such as ToG [35] demonstrated flexibility across multiple reasoning and question-answering paradigms, including slot-filling and fact-checking. Systems using ontology-aligned KG construction achieved high alignment scores (up to 91.5%) and successful conceptual query coverage [19]. Fused embedding-RAG models reported F1-scores of 0.70 with significant gains over baseline models [22], while Cypher-enhanced Q/A systems improved intent discovery and semantic policy mapping [28]. Graph-based systems such as GRAG achieved over 170% improvement on benchmarks

like ExplaGraphs [37], and embedding-transfer pipelines like KG-FIT delivered up to 17.7% gains on link predictions [17]. Multi-agent KG-integrated systems enabled complex scientific workflows while outperforming standalone retrieval [21].

3.6. Performance Analysis

To answer research question RQ3.3, we evaluate the results, limitations, and conclusions for each method. Despite the results described in the previous section, limitations were noted across systems. The completeness and expressivity of KGs and ontologies emerged as a recurring bottleneck, as missing edges or underrepresented concepts limited retrieval accuracy [20, 39, 36, 34]. Regardless of how ontologies are utilized, the structure, design, and naming conventions drastically impact how suitable they are for interactions with LLMs. Small differences in the values and order of data being embedded may give drastically different results [44]. Since the property chains found in ontologies serve as the data being embedded, ontology design is particularly important. Hallucinations and path misalignment introduced by LLMs during reasoning processes posed additional challenges [38, 40]. The computational overhead associated with learning retrieval plans, structural parsing, or hybrid pipelines impacted scalability and cost-efficiency [33, 39]. Methods that embed the ontological relations require less overhead as the structure is embedded in a vector directly, and can take advantage of hierarchical vector search algorithms [23, 22]. Further challenges included sensitivity to prompt design, especially in semi-automated KG generation pipelines, which suffered from inconsistent outputs and limited compatibility with structured query languages like SPARQL [19, 44]. For instance, DRAGON-AI required prompt engineering fixes to avoid retrieval of unrelated biomedical terms due to ambiguous embeddings [23]. In some cases DRAGON-AI would "hallucinate" parent terms that do not exist that are good candidates for inclusion in the ontology. Systems built on low-resource languages reported misclassified clusters due to under performance of NLP tools [28]. Graph retrieval systems were constrained by their reliance on accurate subgraph pruning and alignment with soft LLM prompts [37], and knowledge transfer models such as KG-FIT struggled with highly lexical datasets and lacked refinement flexibility [17]. Ontological graph generation processes continued to be prone to hallucination without safeguards [21].

The conclusions drawn from these works demonstrate the value of integrating structured knowledge into RAG systems. Several studies advocate for refining and expanding domain ontologies to boost coverage and adaptability in diverse application settings [20, 16]. Tools such as DRAGON-AI [23] suggest a future of semi-automated ontology editing supported by LLMs, while bidirectional reasoning strategies and retrieval optimization frameworks like ToG and ORT demonstrate robust pathways forward to improve LLM-based reasoning [36, 35, 38]. Semi-automated KG construction has shown that human-in-the-loop systems remain essential for reliability and scale [19]. The fusion of retrieval and ontology-based embedding has proven to be a scalable and hallucination-resilient solution, particularly in biomedical domains [22]. In cross-domain KG construction, semantic validation techniques continue to improve Q/A performance [31]. Graph retrievers and neural reasoning frameworks such as GRAG support robust multi-hop reasoning without fine-tuning [37], while transfer learning methods such as KG-FIT offer economical embedding refinement [17]. Finally, multi-agent RAG+KG systems can provide a foundation for explainable and domain-adaptable information retrieval in scientific and mission-critical environments [21].

The application of RAG using ontologies and KGs has demonstrated notable advancements in information retrieval, particularly through improved accuracy, multi-hop reasoning, and enriched semantic outputs. Systems that integrated structured knowledge representations showed clear benefits in aligning responses to domain-specific queries, enhancing interpretability, and outperforming traditional RAG baselines on metrics such as F1, Hit@1, and Hits@10. Graph retrievers, ontology-driven Q/A systems, and hybrid models combining pre-trained embeddings with RAG contributed to these improvements across diverse domains, including biomedical data, education, and scientific workflows.

However, these results also reveal several critical limitations. Many systems were highly sensitive to prompt phrasing, which could significantly affect the quality of generated outputs [19]. Ontology and KG completeness were recurring issues, as missing or underrepresented concepts limited retrieval

effectiveness [20, 39, 36, 34]. LLM-generated content often introduced hallucinations or semantic drift, particularly during graph construction or reasoning steps [38, 40, 21]. Some pipelines showed incompatibility with standard representation languages like OWL and querying tools like SPARQL, specifically on their open-world assumption. Foe example, when SPARQL returned an empty result set, implicit semantics (such as implicit subclasses) were reinterpreted as closed-world, resulting in inferred class disjointness [31]. In low-resource language settings, under performance of NLP tools led to duplicated or misclassified outputs [28], while embedding-transfer models struggled on highly lexical datasets and lacked expressive refinement capabilities [17]. Furthermore, although embedding-based approaches reduced computational overhead, they sometimes did so at the expense of query flexibility and contextual depth [22, 23]. These issues collectively suggest that while RAG systems integrated with structured knowledge are advancing, they still face significant challenges in robustness, consistency, and generalizability.

4. Conclusion

This literature review explored the integration of ontologies and knowledge graphs (KGs) with retrieval-augmented generation (RAG) techniques, highlighting their significant potential in enhancing semantic coherence, retrieval accuracy, and multi-hop reasoning. The methodologies reviewed demonstrate benefits across diverse applications, including scientific and engineering workflows, biomedical contexts, education, enterprise data access, media summarization, social sciences, structured professional workflows, and multi-hop Q/A scenarios.

Structured knowledge representations such as ontologies and KGs substantially contribute to RAG performance by providing essential semantic context, improving the alignment of generated responses, and supporting complex reasoning tasks. However, several challenges persist, notably those related to ontology completeness, structural expressivity, and computational efficiency, which can limit the practical applications. For example, property chain embeddings are too unstable to generate queries or compatible graph structures, especially for complex semantics as the approaches do not scale well. Embeddings and especially embedding-transfer, lack refinement capabilities of highly-lexical datasets to ensure better semantic alignment with a given ontology or KG. The sensitivity of current systems to prompt formulation also affects consistency and reliability. Hence, a variety of hallucinations introduced by LLMs during graph generation and reasoning highlight areas requiring further research and improvement. Approaches such as semi-automated ontology editing, graph-guided reasoning, and though-of-chain prompting present promising avenues for addressing these issues; however, continued validation and refinement are still needed. Future research focused on these areas will further enhance the reliability, scalability, and practical effectiveness of ontology-enhanced RAG frameworks.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. 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, volume 33, Curran Associates, Inc., 2020, pp. 9459–9474.

[2] K. Guu, Realm: Retrieval-augmented language model pre-training, in: Proceedings of the 37th International Conference on Machine Learning, volume 119, 2020, pp. 3929–3939. URL: https://arxiv.org/abs/2002.08909.

[3] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 874–880. doi:10.18653/v1/2021.eacl-main.74.

[4] V. Karpukhin, Dense passage retrieval for open-domain question answering, in: Proceedings of

the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

[5] S. Gupta, R. Ranjan, S. N. Singh, A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions, arXiv preprint arXiv:2410.12837 (2024).

[6] S. Gupta, R. Ranjan, Evaluation of LLMs Biases Towards Elite Universities: A Persona-Based Exploration, arXiv preprint arXiv:2407.12801 (2024). URL: https://arxiv.org/abs/2407.12801.

[7] D. Chandrasekaran, V. Mago, Evolution of semantic similarity—a survey, ACM Computing Surveys (CSur) 54 (2021) 1–37.

[8] J. Chen, O. Mashkova, F. Zhapa-Camacho, R. Hoehndorf, Y. He, I. Horrocks, Ontology embedding: a survey of methods, applications and resources, IEEE Transactions on Knowledge and Data Engineering (2025).

[9] P. Gupta, M. Jaggi, Obtaining better static word embeddings using contextual embedding models, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 5241–5253. doi:10.18653/v1/2021.acl-long.408.

[10] Q. Liu, J. Lu, G. Zhang, T. Shen, Z. Zhang, H. Huang, Domain-specific meta-embedding with latent semantic structures, Information Sciences 555 (2021) 410–423.

[11] K. Zhou, K. Ethayarajh, D. Jurafsky, Frequency-based distortions in contextualized word embeddings, arXiv preprint arXiv:2104.08465 (2021).

[12] F. Elsafoury, S. R. Wilson, N. Ramzan, A comparative study on word embeddings and social NLP tasks, in: L.-W. Ku, C.-T. Li, Y.-C. Tsai, W.-Y. Wang (Eds.), Proceedings of the Tenth International Workshop on Natural Language Processing for Social Media, Association for Computational Linguistics, Seattle, Washington, 2022, pp. 55–64. doi:10.18653/v1/2022.socialnlp-1.5.

[13] Z. Rahimi, M. M. Homayounpour, The impact of preprocessing on word embedding quality: A comparative study, Language Resources and Evaluation 57 (2023) 257–291.

[14] Y. Wang, Y. Hou, W. Che, T. Liu, From static to dynamic word representations: a survey, International Journal of Machine Learning and Cybernetics 11 (2020) 1611–1630.

[15] S. Khosla, S. Jain, M. Anupama, R. S. R. Thavva, Comparative analysis of multiple embedding models for text based document similarity, in: International Conference on Artificial Intelligence and Speech Technology, Springer, 2025, pp. 169–180.

[16] K. Sharma, P. Kumar, Y. Li, Og-rag: Ontology-grounded retrieval-augmented generation for large language models, arXiv preprint arXiv:2412.15235 (2024).

[17] P. Jiang, L. Cao, C. D. Xiao, P. Bhatia, J. Sun, J. Han, Kg-fit: Knowledge graph fine-tuning upon open-world knowledge, Advances in Neural Information Processing Systems 37 (2024).

[18] S. Liu, Y. Chen, X. Xie, J. Siow, Y. Liu, Retrieval-augmented generation for code summarization via hybrid gnn, arXiv preprint arXiv:2006.05405 (2020).

[19] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An LLM supported approach to ontology and knowledge graph construction, arXiv preprint arXiv:2403.08345 (2024).

[20] Y. Park, P. Witherell, N. A. Surovi, H. Cho, Ontology-based retrieval augmented generation (rag) for genai-supported additive manufacturing, in: Proceedings of the 35th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference, University of Texas at Austin, 2024, pp. 1587–1600.

[21] M. J. Buehler, Generative retrieval-augmented ontologic graph and multiagent strategies for interpretive large language model-based materials design, ACS Engineering Au 4 (2024) 241–277.

[22] A. Albayrak, Y. Xiao, P. Mukherjee, S. S. Barnett, C. A. Marcou, S. N. Hart, Enhancing human phenotype ontology term extraction through synthetic case reports and embedding-based retrieval: A novel approach for improved biomedical data annotation, Journal of Pathology Informatics 16 (2025) 100409.

[23] S. Toro, A. V. Anagnostopoulos, S. M. Bello, K. Blumberg, R. Cameron, L. Carmody, A. D. Diehl, D. M. Dooley, W. D. Duncan, P. Fey, et al., Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai), Journal of Biomedical Semantics 15 (2024) 19.

[24] W. D. Duncan, T. Thyvalikakath, M. Haendel, C. Torniai, P. Hernandez, M. Song, A. Acharya, D. J. Caplan, T. Schleyer, A. Ruttenberg, Structuring, reuse and analysis of electronic dental data using the oral health and disease ontology, Journal of Biomedical Semantics 11 (2020) 8.
[25] The Dublin CoreTM Metadata Initiative (DCMI), DCMI metadata terms, https://www.dublincore.org/specifications/dublin-core/dcmi-terms/, 2024. Accessed: April 14 2024.
[26] Prov W3C Working Group, Prov-overview, https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/, 2015. Accessed: May 8, 2023.
[27] M. DeBellis, N. Dutta, J. Gino, A. Balaji, Integrating ontologies and large language models to implement retrieval augmented generation, Applied Ontology 19 (2024) 389–407.
[28] T. Bui, O. Tran, P. Nguyen, B. Ho, L. Nguyen, T. Bui, T. Quan, Cross-data knowledge graph construction for LLM-enabled educational question-answering system: a case study at HCMUT, in: Proceedings of the 1st ACM Workshop on AI-Powered Q&A Systems for Multimedia, 2024, pp. 36–43.
[29] G. Schreiber, M. Dean, OWL Web Ontology Language Reference, 2004. URL: https://www.w3.org/TR/owl-ref/.
[30] E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF, 2008. URL: http://www.w3.org/TR/rdf-sparql-query/.
[31] D. Allemang, J. Sequeda, Increasing the LLM accuracy for question answering: Ontologies to the rescue!, arXiv preprint arXiv:2405.11706 (2024).
[32] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, From local to global: A graph rag approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).
[33] X. Zhu, X. Guo, S. Cao, S. Li, J. Gong, StructuGraphRAG: Structured document-informed knowledge graphs for retrieval-augmented generation, Proceedings of the AAAI Symposium Series 4 (2024) 242–251. URL: https://ojs.aaai.org/index.php/AAAI-SS/article/view/31798. doi:10.1609/aaaiss.v4i1.31798.
[34] L. Luo, Y.-F. Li, G. Haffari, S. Pan, Reasoning on graphs: Faithful and interpretable large language model reasoning, arXiv preprint arXiv:2310.01061 (2023).
[35] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. M. Ni, H.-Y. Shum, J. Guo, Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph, arXiv preprint arXiv:2307.07697 (2023).
[36] G. He, Y. Lan, J. Jiang, W. X. Zhao, J.-R. Wen, Improving multi-hop knowledge base question answering by learning intermediate supervision signals, in: Proceedings of the 14th ACM international conference on web search and data mining, 2021, pp. 553–561.
[37] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, L. Zhao, Grag: Graph retrieval-augmented generation, arXiv preprint arXiv:2405.16506 (2024).
[38] R. Liu, B. Luo, J. Li, B. Wang, M. Liu, D. Wu, S. Wang, B. Qin, Ontology-guided reverse thinking makes large language models stronger on knowledge graph question answering, arXiv preprint arXiv:2502.11491 (2025).
[39] J. Wang, M. Chen, B. Hu, D. Yang, Z. Liu, Y. Shen, P. Wei, Z. Zhang, J. Gu, J. Zhou, et al., Learning to plan for retrieval-augmented large language models from knowledge graphs, arXiv preprint arXiv:2406.14282 (2024).
[40] Z. Chen, L. Bai, Z. Li, Z. Huang, X. Jin, Y. Dou, A new pipeline for knowledge graph reasoning enhanced by large language models without fine-tuning, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 1366–1381.
[41] 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.
[42] J. Portisch, N. Heist, H. Paulheim, Knowledge graph embedding for data mining vs. knowledge graph embedding for link prediction–two sides of the same coin?, Semantic Web 13 (2022) 399–422.
[43] OpenAI, Chatgpt (march 14 version), https://chat.openai.com/chat, 2023.
[44] Y. Lu, M. Bartolo, A. Moore, S. Riedel, P. Stenetorp, Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, arXiv preprint arXiv:2104.08786 (2021).