Proceedings of the 5th International Conference on Pervasive Computing and Social Networking (ICPCSN-2025)
IEEE Xplore Part Number: CFP25EZ5-ART; ISBN: 979-8-3315-3519-3

2025 5th International Conference on Pervasive Computing and Social Networking (ICPCSN) | 979-8-3315-3519-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICPCSN65854.2025.11035854

A Knowledge Graph-based RAG for Cross-Document Information Extraction

1st Sudhanshu Suryawanshi
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology & Management
Thane, India
sudhanshu19102003@gmail.com

2nd Shreyas Waghmode
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology & Management
Thane, India
shreyasw469@gmail.com

3rd Ritesh Sawant
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology & Management
Thane, India
riteshsawant145@gmail.com

4th Dr. Megha Gupta
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology & Management
Thane, India
meghagupta@nhitm.ac.in

***Abstract*—Extracting information from multiple complex documents, such as research papers is challenging due to LLMs' limited context size and their difficulty linking distant concepts. To address this, A propose a Knowledge-Graph-based Retrieval-Augmented Generation (RAG) system that builds a knowledge graph from research papers and uses task-specific retrieval methods—combining vector and graph search. This approach not only enhances retrieval but also provides a structured, interpretable representation of information. Our results show that the structured search method outperforms traditional techniques in managing cross-document context, leading to more effective and detailed extraction, and ultimately aiding the comprehension of complex documents.**

***Index Terms*—Knowledge Graph, Retrieval-Augmented Generation, Large Language Models, Vector Search, Graph Search, Information Extraction, Cross-Document Context, Structured Representation.**

I. Introduction

With the increasing complexity and volume of textual data, such as research papers, technical reports, and academic articles, efficiently extracting and synthesizing meaningful information has become a significant challenge. Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language processing tasks like Information Extraction and Summarization [1], [2]; however, their limited context window and difficulty in linking Cross-Document concepts restrict their effectiveness in processing long-form texts. Direct retrieval techniques often fail to provide meaningful insights across multiple documents [3]

To address these challenges, developing a Retrieval Augmented Generation (RAG)-based system that leverages graph methods to enhance information retrieval and summarization. This integration enhances contextual relevance by leveraging both semantic and structural contexts, resulting in more accurate and relevant responses. The use of Retrieval Augmented Generation (RAG) helps mitigate hallucination, providing guardrails that ensure generated responses are grounded in accurate information [4], [5] . Furthermore, the graph-aided system excels in explainability, enabling users to trace the reasoning behind generated outputs through sub-graph matching and semantic similarity, thereby enhancing trust and usability.

A key aspect of our approach is the creation of a domain-specific knowledge graph that organizes the concepts, methodologies, and relationships extracted into a structured format. This graph not only improves the accuracy of the retrieval, but also provides a more interpretable framework for research findings, allowing users to quickly navigate relevant methodologies, compare technologies, and identify optimal approaches within a specific domain. By integrating structured knowledge graphs with vector search, our system enhances the ability to answer queries effectively.

II. Related Work

Self-RAG is an advanced system that improves how large language models (LLMs) work by using retrieval and self-reflection [6]. Adaptive-RAG is a flexible retrieval system that changes its approach based on how difficult a question is. However, its accuracy depends on the quality of the information it retrieves.

The integration of Large Language Models (LLMs) with Knowledge Graphs (KGs) has emerged as a powerful approach for structured knowledge extraction, intelligent query reasoning, and retrieval-augmented systems. Recent studies have explored how LLMs can be leveraged for both automated KG construction and graph-assisted reasoning, improving information retrieval, knowledge organization, and contextual understanding.

LLM4EduKG, introduced by Sun et al. (2024), focuses on the automated construction of educational knowledge graphs (EduKGs) [7]. Traditional EduKG construction relies on manual annotation or deep learning models, both of which pose

scalability and labor challenges. LLM4EduKG addresses this by using a structured prompting framework to extract and evaluate educational triples from heterogeneous textual sources. Experimental results demonstrate that structured prompts improve knowledge extraction accuracy and adaptability, particularly in Chinese-language educational contexts. This approach significantly reduces reliance on manually curated datasets while maintaining high precision.

KG-BERT, proposed by Yao et al. (2019), introduces a transformer-based approach for knowledge graph completion by leveraging pre-trained language models [8]. Traditional knowledge graph completion methods rely on embedding-based techniques, which struggle with complex relational reasoning and unseen entities. KG-BERT addresses these limitations by formulating knowledge graph triples as natural language sentences and applying BERT to predict missing links. Experimental results show that KG-BERT outperforms traditional embedding models on benchmark datasets, demonstrating improved generalization and contextual understanding. This approach highlights the potential of leveraging large language models for more accurate and interpretable knowledge graph reasoning.

Beyond education-focused KGs, GraphAide, introduced by Purohit et al. (2024), enhances graph-assisted query reasoning by integrating retrieval-augmented generation (RAG) with semantic web technologies [9]. Unlike standard RAG systems that depend solely on vector search, GraphAide utilizes subgraph matching and ontology-guided KG construction to refine query accuracy and improve explainability. The system incorporates named entity recognition (NER), entity disambiguation, and multi-modal retrieval, ensuring contextually enriched and precise responses. By grounding LLM outputs in structured knowledge, GraphAide significantly reduces hallucinations, making it well-suited for scientific research, intelligence analysis, and domain-specific QA.

Expanding on these advancements, Knollmeyer et al. (2024) propose an enhanced RAG system that integrates a Document Knowledge Graph (DKG) to improve document-based question answering [10]. Traditional vector-based retrieval systems struggle with metadata handling, structural relationships, and cross-document dependencies. Their approach constructs a DKG using ontologies like the Document Components Ontology (DoCO) to represent document structure, metadata, and hierarchical relationships. During retrieval, the vector database fetches relevant text chunks, and the DKG expands the results by incorporating context-aware metadata and hierarchical relationships. This method enhances semantic coherence in LLM responses, particularly in enterprise knowledge management and factory planning domains.

III. Methodology

A. Document Loading and Chunking

We took a sample of 10 research papers in PDF format and converted them into text using LangChain's [11] *"PyPDFLoader"* class. Then, a class was applied *"RecursiveCharacterTextSplitter"* to split the text into manageable chunks for the LLM to process. A chunk size of 2560 with an overlap of 250 was selected for our experiment. This value was found through experimentation and determined it was sufficient to maintain the coherence of the content. The chunk size should align with the model's context length and output length. If the chunk size is too small, it may lead to a loss of contextual meaning across splits. Conversely, if it is too large, it could exceed the model's context window, leading to truncation or processing inefficiencies.

B. Knowledge graph Construction with Enhancement for Vector Searchability

1) Entity Recognition: In this step, first identify the types of entities relevant to the task. Since the processing had research papers, the focus was on entities such as concepts, topics, persons, papers, models, methods, equations, and variables. These types of entities were chosen based on the structure of research papers and their importance in constructing a meaningful knowledge graph. Next, prompt the LLM to find and categorize these entities using predefined types.

In the next step, add a brief description to the entities. The integration of descriptions facilitates more accurate similarity assessments by aligning nodes with their intended roles and related concepts. For example, when searching for entities like "apple," having a description such as "a fruit" enables the system to understand broader categories, thereby expanding retrieval results in a meaningful way. This contextual understanding is particularly valuable in hybrid search scenarios, where diverse query types or sources need to be aligned cohesively.

Moreover, these descriptions act as a bridge between different data sources and queries, ensuring that searches across various inputs remain coherent and relevant. By providing consistent and meaningful labels, the system can interpret and match queries more effectively, thereby enhancing overall retrieval accuracy and relevance.

2) Relation Extraction:: In this step, extract relationships between the identified entities and complete the graph by categorizing them into specific types such as Co-author, Affiliation, Citation, Result, Methodology, Funding, Research Area, Tool/Resource, etc. These entity types are chosen based on the structure and key components of research papers, ensuring that the extracted relationships accurately represent academic connections. Additionally, provide a detailed description of these relationships, which enhances the ability to retrieve relevant information from the graph. By structuring the data in this way, which create a more meaningful and organized representation of research content.

Unlike standard Named Entity Recognition (NER) approaches, which typically classify entities into broad categories like Person, Organization, and Location, our method is tailored to academic research. This domain-specific approach allows us to capture nuanced relationships, such as collaborations (Co-author, Affiliation), contributions to scientific knowledge (Methodology, Result, Citation). By focusing on these

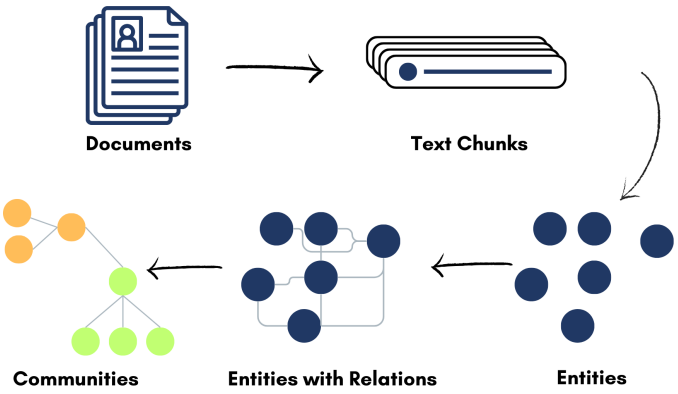

Fig. 1. Knowledge Graph Creation.

research-specific entity types, which improve the relevance of information retrieved, making the graph more useful for analyzing academic literature.

C. Community detection

Community detection algorithms are applied to knowledge graphs to identify clusters of related entities, revealing hidden structures and improving information retrieval. They enhance reasoning by inferring missing links, aid in anomaly detection by spotting outliers, and optimize graph processing by partitioning the graph into meaningful sub-graphs. These algorithms help in organizing complex data, making it suitable for retrieval because it becomes easier to divide-and-conquer global summarization. For our experiment, The widely used Louvain method was employed due to its simplicity and computational efficiency. However, more advanced algorithms can be used for improved accuracy and finer-grained community detection [12].

D. Hierarchical Knowledge Graph Summarization

Knowledge graphs contain vast amounts of interconnected data, necessitating efficient extraction and structuring of relevant information. Our hierarchical approach to knowledge graph summarization captures core topics by identifying key nodes based on their degree, extracting a relevant sub-graph, and organizing related entities into meaningful subtopics. These subtopics form the foundation for structured summaries that provide a concise yet informative representation of the original graph. However, our current approach has certain limitations, such as a restricted search depth of three and node selection criteria that may impact the completeness of the summary. This initial implementation is a foundational step that requires further refinement and optimization in future iterations.

1) Graph Processing: Our methodology follows a structured approach to extract and summarize relevant information from a knowledge graph. The key steps in this process are:

- **Node Selection:** identify the central node with the highest number of connections to serve as the focal point of our analysis. This ensures that our summary captures the most relevant and influential information
- **Sub-graph Extraction:** To narrow down the search space the sub-graph was extract
- **Entity Filtering:** To improve relevance, remove nodes that do not contribute to the conceptual understanding of the graph, such as names of individuals, organizations, and locations.

2) Subtopic Identification and Structuring: To create a structured summary, organize the child nodes of the root node into meaningful subtopics through neighborhood analysis. Each node's connections are examined to determine their contribution to the overarching topic, grouping those with similar contextual relevance. Using an LLM, subtopics where generate based on extracted relationships, ensuring each one captures a distinct aspect of the main topic. These subtopics are then arranged in a logical sequence to enhance readability and comprehension.

3) Information Compilation:: Once the subtopics are identified and structured, The relevant information from the graph is compiled to generate a well-organized summary.

- **Data Extraction:** The detailed information about each node was retrieve by analyzing all its relationships throughout the graph, ensuring a comprehensive understanding of its context.
- **Node Summarization:** The extracted data are transformed into concise summaries for each node, preserving key details while maintaining clarity and relevance.
- **Subtopic Aggregation:** The individual node summaries are grouped based on their assigned subtopics and formatted cohesively.
- **Final Compilation:** The structured subtopic summaries are merged into a single document to create a comprehensive and well-organized output that effectively conveys insights from the knowledge graph.

E. Vector Search-Aided Retrieval

We retrieve information by integrating keyword-based techniques with vector search, ensuring precise and context-aware extraction of relevant entities from the knowledge graph. This is done to answer specific questions or extract targeted information from the graph.

- **Keyword Extraction:** A large language model (LLM) extracts essential keywords from the user query, filtering out stopwords, handel short forms and identifying key terms to optimize the search process.
- **Vector Search:** Once keywords are extract from the user's query, The vector database was searched where knowledge graph entities—stored as numerical embeddings with a name, type, and description—are retrieved and ranked based on their semantic similarity to the query.
- **Graph-Based Retrieval:** The retrieved entities are mapped back to the knowledge graph, allowing relationships with neighboring nodes to be extracted. This step enhances contextual relevance and provides structured insights.

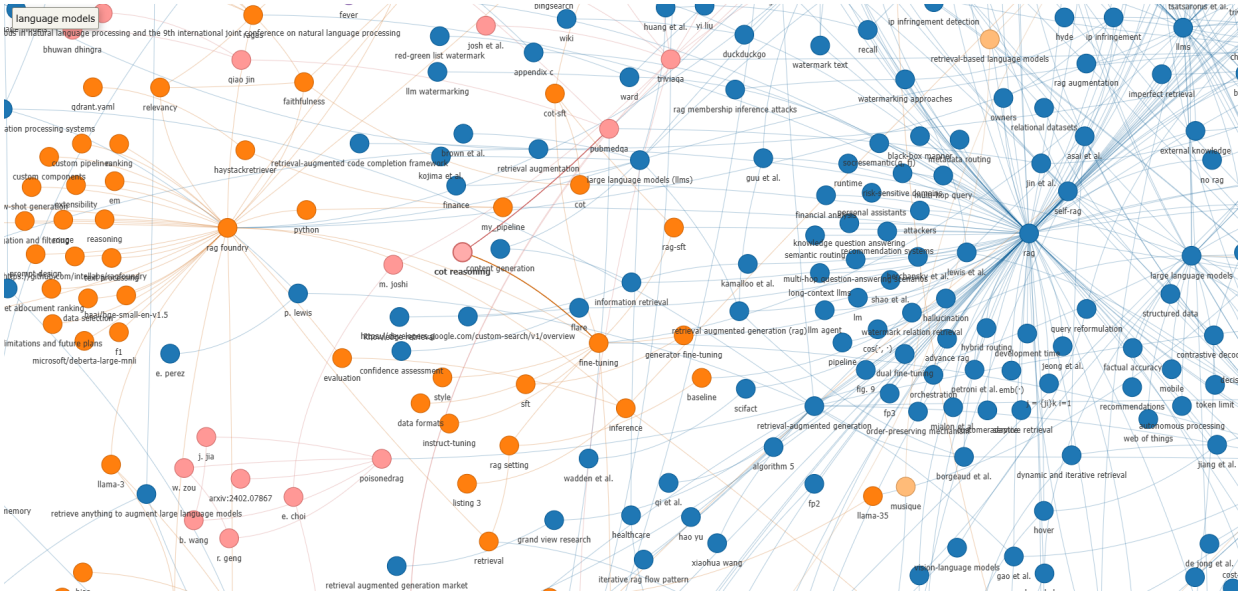

Fig. 2. Knowledge graph visualization.

- **Contextual Summarization:** A large language model (LLM) synthesizes the retrieved knowledge into a coherent response that aligns with the original query.

By leveraging vector search as the foundation of retrieval while incorporating structured knowledge graph relationships, our approach balances precision with comprehensive contextual understanding.

IV. Results

The analysis of the knowledge graph provided insight into its structural properties and community formations. Evaluating degree distribution and relational characteristics revealed key entities and their connectivity patterns. Highly connected nodes serve as hubs, while well-defined communities group related entities, contributing to a better understanding of the graph's organization. Figure 2 presents a visualization of the knowledge graph, showcasing the overall structure, node connectivity, and the formation of distinct communities. This visualization helps in identifying key entities and understanding their relational significance within the graph.

The Table I indicate a well-connected network where a vast number of entities interact through numerous relations. This structure is organized into many distinct communities, most of which tend to be small and tightly knit. However, the presence of a few substantially larger communities suggests that while most clusters represent focused subgroups, there are also some broader, more influential clusters that likely serve as central hubs within the network.

The distribution of node types in Figure 3 shows that *People* nodes dominate due to frequent references in research papers, followed by *Concept*, *Model*, *Paper*, *Topic*, *Method*, *Variables*, *Dataset*, and *Equations*, with *Variables* exceeding *Equations* since equations typically contain multiple variables. In the Figure 4 shows that the most common relationships, including

TABLE I
Graph Analysis Metrics

Metric	Value
Total Entities	2,601
Total Relations	3,052
Number of Communities	311
Average Nodes per Community	8.36
Maximum Nodes in a Community	172
Minimum Nodes in a Community	2

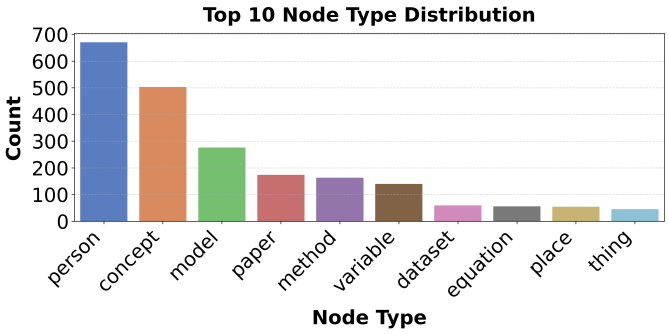

Fig. 3. Distribution of Node Types.

Venue/Publication, *Author_Role*, *Methodology*, and *Tools/Resources*, again highlight the structured nature of citations and methodological references in academic literature. These node types and relationships, heavily referenced in research, emphasize how knowledge is interconnected, providing insight into scholarly contributions and their contextual links.

Figure 5 illustrates the Top 10 Node Degree Distribution in the knowledge graph. The "rag (rag)" node has the highest degree, which aligns with the fact that RAG (Retrieval-

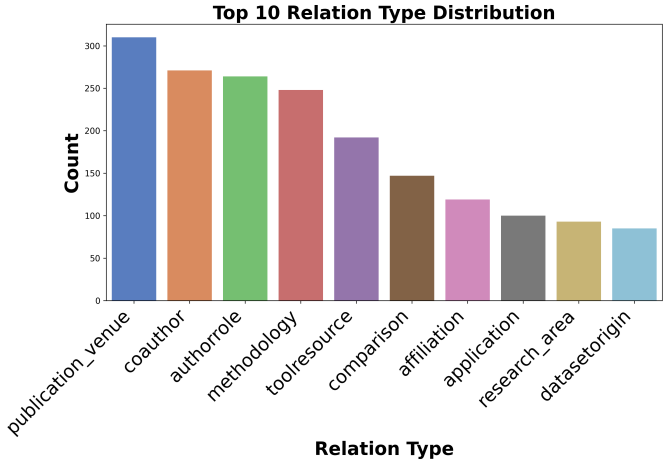

Fig. 4. Distribution of Relation Types.

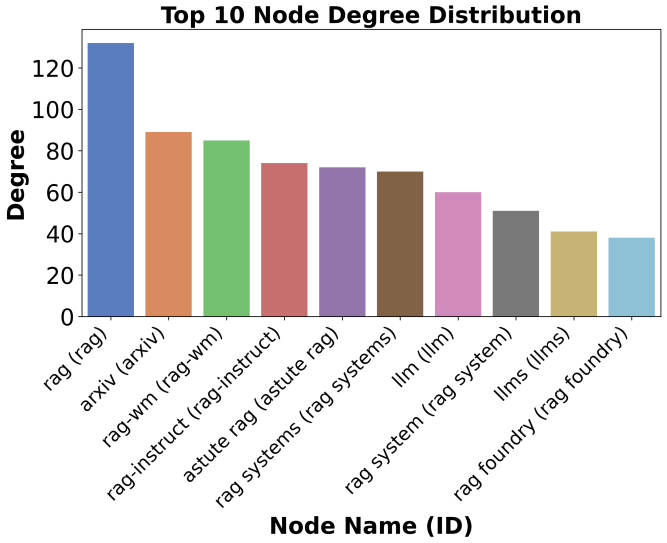

Fig. 5. Top 10 Nodes by Degree.

Augmented Generation) was the primary theme of the input data. The ”arXiv (arXiv)” node appears prominently due to the dataset being sourced from arXiv, leading to frequent references to research papers from the platform. Other high-degree nodes, such as ”rag-wfm,” ”rag-instruct,” and ”rag systems,” indicate that different RAG-related methodologies and systems were widely referenced. This distribution highlights the structured nature of the dataset, where RAG-related entities form major connectivity hubs, emphasizing their importance in the analyzed domain. However, inconsistencies in entity recognition were observed, where semantically equivalent terms (e.g., *“Retrieval-Augmented Generation”* and *“RAG”*, *“LLM”* and *“LLMs”*) were classified as distinct entities, indicating a need for improved entity normalization techniques to ensure consistency in knowledge representation.

Building on this knowledge graph, *Hierarchical Knowledge Graph Summarization* was utilized to generate a comprehensive summary of the research papers. The summarization process resulted in a document containing 12,356 words and 85,910 characters, effectively capturing various subtopics discussed in the input papers while demonstrating strong cross-document contextualization. However, it was observed that when certain subtopics contained an extensive amount of information, the depth of coverage for each topic was significantly diminished. To address this issue, the approach was refined by instructing the model to avoid selecting overly generic subtopics. The provision of explicit examples of relevant subtopics was found to be the most effective strategy in ensuring a well-balanced summary.

Complementing the knowledge graph summarization, our *Vector Search-Aided Retrieval* approach further enhanced query answering and information extraction. The system consistently produced well-structured, comprehensive responses, leveraging a diverse set of generated keywords to optimize vector search. This keyword expansion enabled the retrieval system to identify the most relevant entities within the knowledge graph, leading to the construction of refined sub-graphs. These sub-graphs not only improved retrieval accuracy but also enriched responses with deeper, contextually relevant insights, ultimately ensuring more precise and informative outputs.

V. Conclusion

This study proposed a Knowledge Graph-Based Retrieval-Augmented Generation (RAG) system for cross-document information extraction, enhancing retrieval efficiency, contextual relevance, and interpret-ability through structured knowledge graphs. By integrating graph-based retrieval with vector search for entity selection, the system demonstrated improved accuracy and organization of research findings. Hierarchical summarization effectively captured core topics, while community detection refined knowledge structuring. However, challenges such as entity recognition inconsistencies, search depth constraints, and rudimentary node selection remain. Future advancements may focus on optimizing subtopic selection, improving scalability, and incorporating adaptive retrieval mechanisms to enhance structured retrieval and knowledge representation for more reliable LLM-driven information extraction.

Acknowledgment

The authors acknowledge the use of large language models like GPTs, for assisting in language refinement and formatting improvements in this paper. However, all technical content, analysis, and conclusions are solely the authors’ responsibility.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[2] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, “Benchmarking large language models for news summarization,” 2023. [Online]. Available: https://arxiv.org/abs/2301.13848

[3] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," 2023. [Online]. Available: https://arxiv.org/abs/2211.08411
[4] K. Bhushan, Y. Nandwani, D. Khandelwal, S. Gupta, G. Pandey, D. Raghu, and S. Joshi, "Systematic knowledge injection into large language models via diverse augmentation for domain-specific rag," 2025. [Online]. Available: https://arxiv.org/abs/2502.08356
[5] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, and T. Derr, "Knowledge graph prompting for multi-document question answering," 2023. [Online]. Available: https://arxiv.org/abs/2308.11730
[6] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," 2023. [Online]. Available: https://arxiv.org/abs/2310.11511
[7] J. Sun, Z. Zhang, and X. He, "Llm4edukg: Llm for automatic construction of educational knowledge graph," in *2024 International Conference on Networking and Network Applications (NaNA)*, 2024, pp. 269–275.
[8] M. Trajanoska, R. Stojanov, and D. Trajanov, "Enhancing knowledge graph construction using large language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.04676
[9] S. Purohit, G. Chin, P. S. Mackey, and J. A. Cottam, "Graphaide: Advanced graph-assisted query and reasoning system," in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 3485–3493.
[10] S. Knollmeyer, M. U. Akmal, L. Koval, S. Asif, S. G. Mathias, and D. Großmann, "Document knowledge graph to enhance question answering with retrieval augmented generation," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2024, pp. 1–4.
[11] H. Chase, "Langchain," 2022, version 1.2.0, released on 2022-10-17. [Online]. Available: https://github.com/langchain-ai/langchain
[12] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," 2024. [Online]. Available: https://arxiv.org/abs/2404.16130