Knowledge Graph Based Retrieval-Augmented Generation for Multi-Hop Question Answering Enhancement

Mahdi Amiri Shavaki
School of Electrical and Computer Engineering, College of Engineering, University of Tehran,
Tehran, Iran
mahdiamiri@ut.ac.ir

Pouria Omrani
Faculty of Electrical Engineering, K. N. Toosi University of Technology,
Tehran, Iran
pouria.omrani@ieee.org

Ramin Toosi
School of Electrical and Computer Engineering, College of Engineering, University of Tehran,
Tehran, Iran
r.toosi@ut.ac.ir

Mohammad Ali Akhaee
School of Electrical and Computer Engineering, College of Engineering, University of Tehran,
Tehran, Iran
akhaee@ut.ac.ir

2024 15th International Conference on Information and Knowledge Technology (IKT) | 979-8-3315-2225-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/IKT65497.2024.10892619

***Abstract*—Multi-hop question answering (QA), which requires integrating information from multiple sources, poses significant challenges in natural language processing. Existing methods often struggle with effective retrieval across documents, leading to incomplete or inaccurate answers. Building upon Graph-based Retrieval-Augmented Generation (Graph RAG), we enhance multi-hop QA by leveraging structured knowledge graphs. Specifically, we construct individual knowledge graphs for each document, where entities are represented as nodes and the relationships between them as edges enriched with contextual properties. These individual graphs are then seamlessly integrated into a comprehensive, unified graph that captures cross-document relationships. Our method improves retrieval by utilizing vector embeddings of these graph relations, enabling more effective multi-hop reasoning across the interconnected data. To evaluate our approach, we assembled a dataset of 500 documents paired with 296 multi-hop questions requiring cross-document information retrieval. Our contributions include developing a novel graph-based retrieval mechanism that leverages vector embeddings of graph relations within the Graph RAG framework, and assembling a comprehensive dataset for multi-hop QA. Comparative experiments show that our enhanced Graph RAG method significantly outperforms the baseline in factual accuracy and semantic similarity, as measured by the RAGAS framework. Additionally, an LLM-based evaluator highlights our method's superior performance in answer comprehensiveness, empowerment, and directness.** [1]

[1]The source code and dataset are available at: https://github.com/AmiriShavaki/KG-based-RAG-for-Multi-hop-QA

***Keywords*—RAG, Graph RAG, Generative AI, LLM, Multi-hop QA, NLP**

I. Introduction and Related Works

Natural Language Processing (NLP) has become a cornerstone of artificial intelligence, enabling machines to understand, interpret, and generate human language. Its applications are vast and impactful, ranging from machine translation [1], text classification [2], sentiment analysis [3], to information retrieval [4]. These applications have transformed industries by enhancing communication between humans and machines, improving data analysis, and facilitating more intuitive user interactions.

LLMs have revolutionized the field of NLP by providing powerful capabilities for understanding and generating text. Models such as GPT-4 [5], Llama-3 [6], and PaLM-2 [7] have been employed in a variety of applications, including text summarization [8], question answering [9], dialogue systems [10] and content creation [11]. The scalability and versatility of LLMs have significantly advanced the state-of-the-art in many NLP tasks, enabling more sophisticated and contextually aware language processing.

RAG is a prominent approach that combines retrieval mechanisms with generative models to enhance the performance of language tasks by grounding the output on external knowledge sources [12]. By integrating retrieval capabilities, RAG methods can access and utilize up-to-date information, making them highly effective for tasks that require factual accuracy and context-awareness [13], [14]. Variations of RAG have been explored to improve different aspects of language processing, such as combining dense and sparse retrieval methods [15], integrating knowledge graphs [16], and optimizing retrieval during

979-8-3315-2225-4/24/$31.00 ©2024 IEEE

training [17]. The importance of RAG lies in its ability to mitigate the limitations of language models, particularly in handling tasks that necessitate accessing and reasoning over external data sources [18]. Recent advancements have focused on enhancing retrieval mechanisms [19], and adapting RAG for specific applications like open-domain question answering [20].

Multi-hop question answering involves addressing complex queries that require reasoning over multiple pieces of information or documents. This task presents significant challenges, as it necessitates the ability to perform multi-step reasoning and integrate information from disparate sources [21]. Approaches to multi-hop question answering include graph-based methods [22], neural reasoning models [23], and iterative retrieval techniques [24]. These methods aim to enhance the system's capability to handle complex queries effectively by building connections between various data points. The MuSiQue dataset was recently introduced to address the challenge of requiring genuine multihop reasoning, with 25K 2-4 hop questions designed to reduce shortcuts and improve model performance on true multihop tasks [25].

Despite the advancements in RAG methods, there remains a significant gap in effectively applying these approaches to multi-hop question answering tasks, where reasoning over multiple pieces of evidence is required. Traditional RAG models often struggle with integrating information from disparate sources to form coherent and accurate answers. To address this limitation, we propose an approach that leverages vector embedding retrieval from a graph relations database. By constructing a knowledge graph from the documents and embedding the relations, our method enhances the retrieval step in RAG, enabling more effective multi-hop reasoning by explicitly modeling the relationships between entities and concepts. This approach facilitates the retrieval of relevant information, thereby improving performance on complex question answering tasks.

In this paper, we present a novel method designed to improve multi-hop question answering by integrating vector embedding retrieval from a graph relations database into the RAG framework. Our approach addresses the limitations of traditional RAG methods in handling multi-hop reasoning by leveraging the structured information of knowledge graphs, enhancing the retrieval process through vector embeddings of graph relations, and meticulously assembling a comprehensive dataset tailored for multi-hop reasoning tasks. This carefully constructed dataset ensures that the knowledge graph accurately represents the relationships necessary for effective multi-hop reasoning.

Thus, our contributions can be listed as:

- **Graph-based Vector Embedding Retrieval**: We develop a novel method that constructs a knowledge graph from the documents, embeds the relations using vector embeddings, and stores them in a vector database to enable efficient retrieval. In our knowledge graph, entities extracted from the documents are represented as nodes, and the relationships between them are depicted as edges enriched with contextual properties such as temporal data and attributes. This structured representation allows for capturing the semantic nuances of the data, facilitating more accurate and contextually relevant retrieval.
- **Improved Multi-hop Question Answering**: By integrating graph-based vector embedding retrieval into the RAG framework, we achieve significant enhancements in multi-hop question answering tasks. Our proposed method outperforms the baseline with a 31.6% increase in average answer correctness (0.649 vs. 0.493) and a 4.75% improvement in average answer similarity (0.948 vs. 0.905).
- **Dataset**: We introduce a comprehensive dataset tailored for multi-hop question answering tasks, encompassing diverse and complex queries that require multi-step reasoning.

The structure of this paper is organized as follows: In Section II, we detail our proposed Graph RAG method, including the dataset preparation, knowledge graph construction, and answer generation process. Section III presents the experimental settings, evaluation metrics, and a comprehensive analysis of the results. Finally, in Section IV, we summarize our findings and discuss potential future work.

II. Proposed Method

A. Dataset

The dataset assembled for this study comprises 500 documents, each consisting of short textual passages ranging from one to several sentences. These documents serve as the primary knowledge sources for answering a set of 296 multi-hop questions specifically designed to require information retrieval across multiple documents. A multi-hop question is defined as one that necessitates the integration of information from more than one source to arrive at a comprehensive answer. In this dataset, no single document contains sufficient information to fully answer any of the questions. Each question, therefore, requires the retrieval of at least two documents.

Of the 500 documents, 100 are real-world texts, sourced from the internet, with references to their original locations. The remaining 400 documents were generated using an LLM and subsequently validated by human supervisors to ensure the quality and relevance of the generated content. Fig. 2 illustrates the composition of the dataset, showing the proportion of real-world texts versus LLM-generated documents.

The dataset was curated with the primary objective of evaluating the performance of our proposed Graph RAG method in comparison to the baseline RAG approach. This dataset serves as a critical resource in the assessment and refinement of our method, which leverages knowledge graphs constructed from individual documents. These

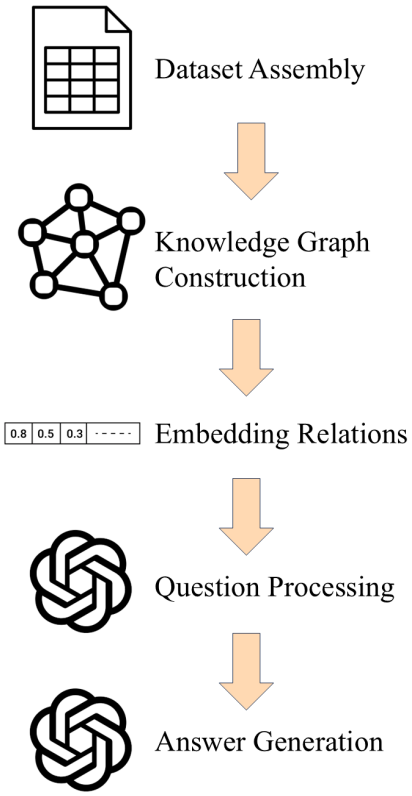

Fig. 1: The overall pipeline of the proposed method consists of five key steps. First, dataset assembly gathers and prepares the relevant data. Second, a knowledge graph construction step is performed to establish relationships between entities. Third, embedding relations transforms the knowledge graph into an embedding vector format, enabling fast nearest vector search among relations. Fourth, question processing involves interpreting the input query in the context of the embedded knowledge graph. Finally, in the fifth step, answer generation produces responses based on the processed query and the embedded relations.

knowledge graphs are subsequently merged to form a comprehensive graph for the entire document set. Relationships within the knowledge graph are embedded and stored in a vector database, allowing for efficient indexing and fast similarity search. The overall process is summarized in Fig. 1.

B. Knowledge Graph Construction

In this study, we construct a knowledge graph from a collection of 500 documents using an LLM to extract meaningful relations between entities. Each document is processed individually to extract relations and entities, which are then represented as graph nodes and edges, respectively. To store and manage the knowledge graph, we utilize Neo4j, a graph database that supports efficient querying and management of complex, interconnected data.

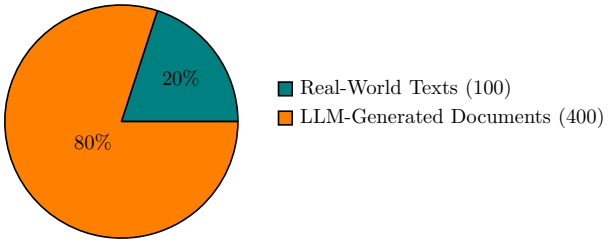

Fig. 2: Composition of the dataset. The pie chart illustrates the distribution of texts within the dataset used for this study. Specifically, 20% (100 texts) are sourced from real-world documents, while 80% (400 texts) are generated by LLMs.

The process begins by parsing each document with a system prompt designed to guide the LLM in extracting relations. This prompt, based on a system developed by [26], was modified to suit the structure and complexity of our dataset. The extracted relations from each document are stored separately, and the individual knowledge graphs are merged to form a comprehensive graph for the entire dataset.

Our experiments revealed that processing each document separately and then merging the resulting graphs produces better results than attempting to generate the entire knowledge graph in a single LLM call or through batched calls. This approach ensures that relations are captured more accurately and contextually, without the risk of missing critical connections. During graph construction, additional information pertaining to the relations (e.g., temporal data, attributes) is stored as properties of the nodes or edges. This richer representation of the data enhances retrieval accuracy during the question-answering process.

In the sample subgraph of Fig. 3, consider the following relations:

- `(Real Sociedad, WON, Copa Del Rey)` with the property `year=2020`
- `(Real Sociedad, SHIRT_COLOUR, Blue And White)`

This structure facilitates answering the question, "What was the primary color of the team that won the Copa del Rey in 2020?" By utilizing the property `year=2020` associated with the `WON` relation, the method identifies `Real Sociedad` as the team that won the `Copa del Rey` in 2020, and according to the `SHIRT_COLOUR` relation, it should determine that the team's primary colors are `Blue And White`. This example demonstrates how incorporating properties into relations within the knowledge graph enhances the system's ability to perform precise multi-hop reasoning in subsequent stages.

C. Answer Generation

Once the knowledge graph is constructed, the next stage involves generating answers to multi-hop questions using

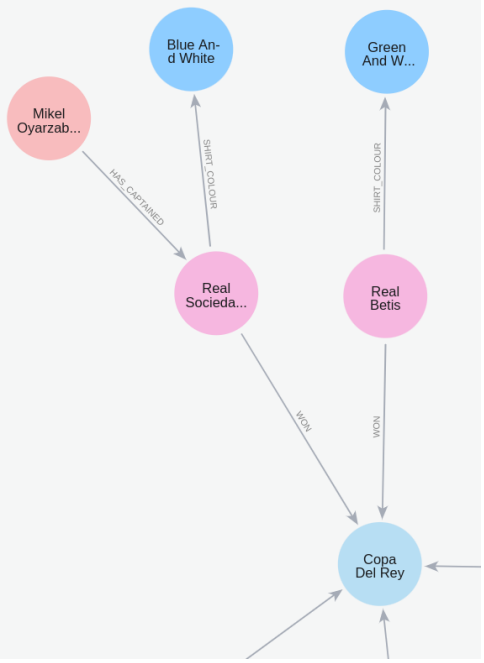

Fig. 3: Sample subgraph of the constructed knowledge graph. This subgraph demonstrates how specific relations and their properties are utilized for multi-hop question answering.

a RAG approach. For the retrieval step, we embed the relations in the knowledge graph into a vector database to enable fast similarity search. A vector database offers efficient retrieval mechanisms by using approximate indexing algorithms based on vector distance metrics, which is essential for handling the large-scale data represented in the knowledge graph. The answer generation workflow is illustrated in Fig. 4.

To create the embeddings, we use the names of the two head entities of each relation, the relation itself, and all associated properties (including key details such as dates and times). This structured embedding process captures the most relevant features of each relation and ensures that important context (especially temporal information) is factored into the answer generation process.

For the retrieval mechanism, we designed a custom prompt in which the user's query is passed to the LLM. The LLM is instructed to identify incomplete relations, where one or more entities or connections may be missing, and suggest which relations to search for in the vector database. This step leverages the LLM's capability to handle incomplete information, and the vector search helps to identify the missing components of the answer.

Once relevant relations are retrieved, a second LLM call is made to generate the answer. In this stage, we explicitly instruct the LLM to limit its response to the knowledge derived from the retrieved relations, ensuring that the generated answer is grounded in the knowledge graph rather than the broader pre-existing knowledge encoded

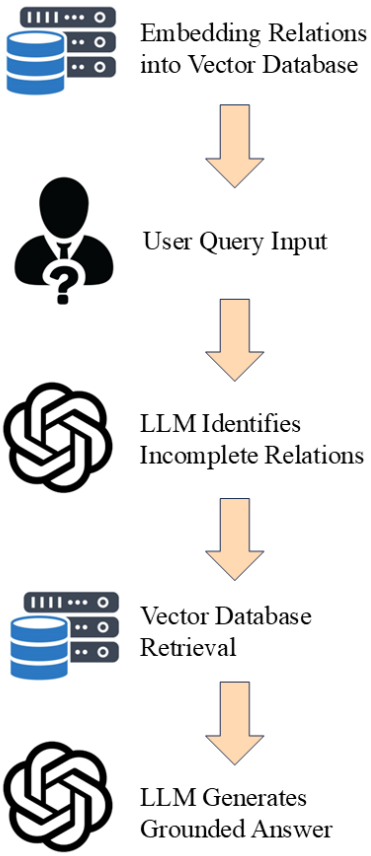

Fig. 4: Answer generation workflow. This figure illustrates the workflow for generating a grounded answer using an LLM and a vector database. First, relations from existing data are embedded into a vector database. When a user submits a query, the LLM analyzes the query to identify any incomplete or missing relations. Next, the vector database retrieves relevant information based on the query, completing the missing relations. Finally, the LLM generates a grounded answer by synthesizing the retrieved information and the user's query. This process ensures that the generated answer is both comprehensive and contextually relevant.

within the LLM itself. This method ensures that the answers remain closely aligned with the available data and the structure of the knowledge graph.

III. Experiments and Results

This section presents the experimental setup, evaluation metrics, and the results obtained from both the proposed method and the baseline RAG approach.

A. Experimental Settings

Our proposed method configurations are outlined in Table I and the baseline RAG approach is configured as detailed in Table II.

Table I: Configuration of the Proposed Method.

Component	Configuration
Graph Database	Neo4j
Vector Database	FAISS
Vector Similarity Metric	Euclidean Distance (L2)
Embeddings	text-embedding-ada-002
Knowledge Graph Construction	GPT-4o
Relation Extraction	GPT-4o
Answer Generation Model	GPT-3.5-turbo

Table II: Configuration of the Baseline RAG Approach.

Component	Configuration
Vector Database	Weaviate
Vector Similarity Metric	Euclidean Distance (L2)
Embeddings	text-embedding-ada-002
Answer Generation Model	GPT-3.5-turbo

In both our proposed method and the baseline RAG approach, we utilize the Euclidean distance (L2 norm) as the similarity metric within the vector databases. The Euclidean distance between two embedding vectors x and y is calculated using the formula:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

where n represents the dimensionality of the embedding vectors, and x_i and y_i are the components of the vectors x and y, respectively. This metric measures the straight-line distance between two points in the embedding space, facilitating the retrieval of semantically similar documents based on their vector representations.

B. Metrics

1) RAGAS: The RAGAS (Retrieval Augmented Generation Assessment) framework [27] was utilized to measure two key aspects: answer similarity and answer correctness.

Answer similarity evaluates the semantic similarity between the generated answers and the ground truth answers. Utilizing the RAGAS framework, we quantified how closely the responses from both the proposed method and the baseline RAG approach aligned with the LLM-generated ground truth.

Answer correctness assesses the factual accuracy of the generated responses by combining factual correctness and semantic similarity between the generated answer and the ground truth. To compute this metric, we prepared a ground truth dataset comprising answers generated by an LLM in response to our dataset's questions. It is important to note that these LLM-generated answers were not constrained to the dataset's knowledge base and could incorporate pre-encoded information. Factual correctness quantifies the factual overlap between the generated answer and the ground truth answer using the concepts of:

- **TP (True Positive):** Facts or statements present in both the ground truth and the generated answer.
- **FP (False Positive):** Facts or statements present in the generated answer but not in the ground truth.
- **FN (False Negative):** Facts or statements present in the ground truth but not in the generated answer.

We compute the F1 Score to measure factual correctness:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Semantic similarity between the generated answer and the ground truth is calculated using cosine similarity of embeddings.

The answer correctness metric is then computed as a weighted sum of the factual correctness and semantic similarity [27]:

$$\begin{aligned} Answer\ Correctness = {} & w_f \times F1\ Score \\ & + w_s \times Semantic\ Similarity \end{aligned} \tag{5}$$

Where w_f and w_s are the weights assigned to factual correctness and semantic similarity, respectively, satisfying $w_f + w_s = 1$. In this study, we set the weight for factual correctness to $w_f = 0.75$ and the weight for semantic similarity to $w_s = 0.25$.

2) LLM Evaluator: In addition to RAGAS, we employed an LLM Evaluator based on a head-to-head comparison methodology, inspired by the approach used in [28]. This evaluator does not require a predefined ground truth, offering a more flexible assessment of answer quality.

The LLM Evaluator operates by presenting an LLM with the question and the corresponding answers from both the proposed method and the baseline RAG approach. The LLM is prompted to determine which of the two answers is superior in terms of Comprehensiveness, Diversity, Empowerment and Directness. This process is repeated for each question in the dataset.

For each question, the evaluator records whether the proposed method's answer is better or worse to the baseline's answer. After processing all questions, we aggregate the results to determine the overall performance comparison between the two methods. This metric provides a direct and qualitative measure of the relative strengths and weaknesses of each approach without relying on an external ground truth.

C. Result Analysis

The RAGAS two key metrics results are summarized in Table III and the number of entries where the proposed method outperformed or underperformed the baseline

Table III: RAGAS Framework Results on Both Datasets. The best result in each metric is marked in bold.

Metric	Our Dataset		Testset of Musique	
	Proposed	**Baseline**	**Proposed**	**Baseline**
Correctness	**0.649**	0.493	**0.477**	0.382
Similarity	**0.948**	0.905	**0.928**	0.876

Table IV: RAGAS Metric Comparison Counts on Both Datasets. The best result in each metric is marked in bold. The columns indicate the number of wins for each method in the metrics: "Proposed" shows how many times the proposed method outperformed the baseline, while "Baseline" indicates how many times the baseline method outperformed the proposed method.

Metric	Our Dataset		Testset of Musique	
	Proposed	**Baseline**	**Proposed**	**Baseline**
Correctness	**186**	110	**17**	8
Similarity	**170**	126	**18**	7

RAG approach are detailed in Table IV. Additionally, the LLM evaluator results are presented in Table V.

The proposed method significantly outperforms the baseline RAG approach in both answer correctness and answer similarity metrics, across both our dataset and the MuSiQue testset. Specifically, on our dataset, the proposed method achieved an average answer correctness score of 0.649, compared to 0.493 for the baseline, indicating a substantial improvement in the factual accuracy of the generated answers. On the MuSiQue testset, the proposed method also showed a marked improvement, achieving a score of 0.477 compared to 0.382 for the baseline. Furthermore, the count of entries where the proposed method outperformed the baseline RAG is notably higher for both metrics on our dataset, and it also demonstrates a stronger performance on the MuSiQue testset, where the proposed method outperformed the baseline in 17 cases compared to 8 for the baseline in answer correctness, and 18 versus 7 in answer similarity. These results collectively highlight the effectiveness of the proposed method in enhancing both the accuracy and the relevance of the generated answers, even on a more challenging benchmark like MuSiQue.

Table V: LLM Evaluator Results on Both Datasets. The best result in each dimension is marked in bold. The columns labeled "P" shows how many times the proposed method outperformed the baseline, while the columns labeled "B" indicates how many times the baseline method outperformed the proposed method.

Dimension	Our Dataset		Testset of Musique	
	P	**B**	**P**	**B**
Comprehensiveness	**224**	72	**20**	5
Diversity	**155**	141	**14**	11
Empowerment	**246**	50	**21**	4
Directness	**162**	134	**16**	9

The LLM evaluator results indicate that the proposed method consistently outperforms the baseline RAG approach across most qualitative dimensions, on both our dataset and the MuSiQue testset. In the Comprehensiveness category, the proposed method was preferred in 224 instances on our dataset compared to 72 for the baseline, and it was also preferred in 20 instances compared to 5 for the baseline on the MuSiQue testset, showcasing its ability to provide more thorough and complete answers. Similarly, in the Empowerment dimension, the proposed method significantly led with 246 wins against 50 for the baseline on our dataset, and with 21 wins against 4 for the baseline on the MuSiQue testset, suggesting that it empowers users with more actionable and relevant information. The Directness metric also favored the proposed method, with 162 wins compared to 134 for the baseline on our dataset, and 16 versus 9 on the MuSiQue testset, indicating that the proposed method generates more straightforward and easily interpretable answers. However, in the Diversity category, the performance was more balanced, with the proposed method winning 155 times and the baseline winning 141 times on our dataset, and 14 versus 11 on the MuSiQue testset. This slight edge suggests that while the proposed method excels in several areas, maintaining diversity in responses remains an area where both methods perform comparably.

Overall, the LLM Evaluator results reinforce the findings from the RAGAS framework, demonstrating that the proposed method not only enhances factual accuracy and similarity but also excels in providing comprehensive, empowering, and direct answers. This is true across both our dataset and the MuSiQue testset, further establishing the proposed method as a superior approach for enhancing multi-hop question answering, even on more challenging and realistic test data.

IV. Conclusion

In this paper, a novel approach to enhance multi-hop question answering was proposed by incorporating knowledge graphs into the RAG framework. Knowledge graphs were constructed for individual documents, and their relations were embedded into a vector database, improving the retrieval and reasoning process. More accurate and contextually relevant answers were generated as a result. Experiments were conducted on a dataset curated specifically for multi-hop reasoning, and it was demonstrated that the proposed method significantly outperformed the baseline RAG model in terms of factual accuracy and answer comprehensiveness, as validated by the RAGAS framework and an LLM-based evaluator.

The results indicate that structured knowledge representations within the RAG framework are highly effective for multi-hop question answering tasks. Limitations of

traditional RAG models in integrating information from multiple sources were addressed through this method. Building on the promising outcomes of this study, future research can explore the extension of this approach to multimodal data, which encompasses text, images, tables, formulas, and other data types. Additionally, incorporating weighted relations within the knowledge graphs presents an avenue for improving retrieval accuracy.

References

[1] H. Wang, H. Wu, Z. He, L. Huang, K. W. Church, Progress in machine translation, Engineering 18 (2022) 143–153.

[2] P. Omrani, Z. Ebrahimian, R. Toosi, M. A. Akhaee, Bilingual covid-19 fake news detection based on lda topic modeling and bert transformer, in: 2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA), 2023, pp. 01–06. doi:10.1109/IPRIA59240.2023.10147179.

[3] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, M. Mridha, Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review, Natural Language Processing Journal (2024) 100059.

[4] K. A. Hambarde, H. Proenca, Information retrieval: recent advances and beyond, IEEE Access (2023).

[5] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[6] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).

[7] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al., Palm 2 technical report, arXiv preprint arXiv:2305.10403 (2023).

[8] H. Jin, Y. Zhang, D. Meng, J. Wang, J. Tan, A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods, arXiv preprint arXiv:2403.02901 (2024).

[9] Y. Zhuang, Y. Yu, K. Wang, H. Sun, C. Zhang, Toolqa: A dataset for llm question answering with external tools, Advances in Neural Information Processing Systems 36 (2023) 50117–50143.

[10] Z. Yi, J. Ouyang, Y. Liu, T. Liao, Z. Xu, Y. Shen, A survey on recent advances in llm-based multi-turn dialogue systems, arXiv preprint arXiv:2402.18013 (2024).

[11] Y. Choi, E. J. Kang, S. Choi, M. K. Lee, J. Kim, Proxona: Leveraging llm-driven personas to enhance creators' understanding of their audience, arXiv preprint arXiv:2408.10937 (2024).

[12] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T.-W. Kuo, N. Guan, et al., Retrieval-augmented generation for natural language processing: A survey, arXiv preprint arXiv:2407.13193 (2024).

[13] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, Q. Li, A survey on rag meeting llms: Towards retrieval-augmented large language models, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6491–6501.

[14] P. Omrani, A. Hosseini, K. Hooshanfar, Z. Ebrahimian, R. Toosi, M. Ali Akhaee, Hybrid retrieval-augmented generation approach for llms query response enhancement, in: 2024 10th International Conference on Web Research (ICWR), 2024, pp. 22–26. doi:10.1109/ICWR61162.2024.10533345.

[15] K. Sawarkar, A. Mangal, S. R. Solanki, Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers, arXiv preprint arXiv:2404.07220 (2024).

[16] D. Sanmartin, Kg-rag: Bridging the gap between knowledge and creativity, arXiv preprint arXiv:2405.12035 (2024).

[17] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, F. Silvestri, The power of noise: Redefining retrieval for rag systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 719–729.

[18] F. Petroni, F. Siciliano, F. Silvestri, G. Trappolini, Ir-rag@ sigir24: Information retrieval's role in rag systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 3036–3039.

[19] Y. Shi, X. Zi, Z. Shi, H. Zhang, Q. Wu, M. Xu, Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems, arXiv preprint arXiv:2407.10670 (2024).

[20] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, S. Nanayakkara, Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering, Transactions of the Association for Computational Linguistics 11 (2023) 1–17.

[21] V. Mavi, A. Jangra, A. Jatowt, A survey on multi-hop question answering and generation, arXiv preprint arXiv:2204.09140 (2022).

[22] S. Mitra, R. Ramnani, S. Sengupta, Constraint-based multi-hop question answering with knowledge graph, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, 2022, pp. 280–288.

[23] Z. Tang, S. Pei, X. Peng, F. Zhuang, X. Zhang, R. Hoehndorf, Neural multi-hop logical query answering with concept-level answers, in: International Semantic Web Conference, Springer, 2023, pp. 522–540.

[24] Z. Jiang, M. Sun, L. Liang, Z. Zhang, Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach, arXiv preprint arXiv:2407.13101 (2024).

[25] H. Trivedi, N. Balasubramanian, T. Khot, A. Sabharwal, musique: Multihop questions via single-hop question composition, Transactions of the Association for Computational Linguistics 10 (2022) 539–554.

[26] T. Bratanic, Constructing knowledge graphs from text using openai functions, bratanic-tomaz.medium.com/constructing-knowledge-graphs-from-text-using-openai-functions-096a6d010c17, accessed: Sep. 28, 2024 (2023).

[27] S. Es, J. James, L. Espinosa-Anke, S. Schockaert, Ragas: Automated evaluation of retrieval augmented generation, arXiv preprint arXiv:2309.15217 (2023).

[28] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, J. Larson, From local to global: A graph rag approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).