**武汉大学本科毕业设计(论文)开题报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | Optimizing Knowledge Representation and Retrieval in RAG Systems Using Hybrid Search and Graph Summarization | | |
| 学生姓名 | Chaiwat Plongkaew | 学号 | 2021326660023 |
| 所在学院 | School of Computer Science | 专业 | Software Engineering |
| 指导教师 | 邓娟 | 职称 | Associate Professor |
| **Objectives and Significance**  The primary objective of this research is to optimize the performance of traditional Retrieval-Augmented Generation (RAG) systems by integrating Knowledge Graphs (KGs) into their workflow. Traditional RAG systems often face limitations such as a lack of understanding of how the broader meanings or contexts of documents connect with one another, hallucinations caused by incomplete or irrelevant retrieved knowledge, and inefficiencies in handling large, complex graph structures, particularly for multi-hop reasoning. These limitations restrict their ability to provide accurate, contextually relevant, and explainable outputs, especially in knowledge-intensive tasks.  The proposed method introduces a hybrid pipeline that integrates Knowledge Graphs (KGs) into traditional Retrieval-Augmented Generation (RAG) systems to address key limitations and enhance their performance. By incorporating structured knowledge, the method enables the system to capture and utilize relationships between the broader meanings and contexts of documents, addressing the lack of understanding of how the overall content of one document connects with another. This is achieved through community detection techniques, which group related entities and documents into meaningful clusters, providing a structured understanding of inter-document relationships and improving the coherence of retrieved and synthesized information. To mitigate hallucinations caused by incomplete or irrelevant retrieved knowledge, the approach leverages the structured nature of KGs as a reliable source of truth, reducing dependence on noisy or ambiguous unstructured data. Additionally, mechanisms such as high-confidence filtering and re-ranking ensure that only the most relevant and contextually appropriate knowledge is utilized, minimizing unsupported outputs. To overcome inefficiencies in handling large, complex graph structures for multi-hop reasoning, the method organizes knowledge into manageable communities using graph-based algorithms, enabling efficient reasoning and systematic synthesis of insights across interconnected data. By addressing these challenges, the proposed method significantly enhances the reliability, accuracy, and scalability of RAG systems for knowledge-intensive tasks.  **Current State of Research and Trends (Domestic and International)**  Research on RAG systems is rapidly evolving both domestically and internationally. In China, the development of Knowledge Graphs has primarily focused on applications in areas such as natural language processing, recommendation systems, and semantic search. However, the integration of KGs with RAG systems for optimizing retrieval pipelines has received limited attention. Current domestic efforts often prioritize constructing large-scale KGs and embedding techniques, but they lack specific implementations in hybrid search contexts.  Internationally, RAG systems have seen significant advancements, particularly in the incorporation of hybrid search strategies that combine vector-based and KG-based approaches. Studies in this area have demonstrated that KGs can enhance retrieval accuracy through entity disambiguation and semantic reasoning. For example, hybrid search methods have been used to integrate graph embeddings with vector models, resulting in improved performance on complex queries. A notable trend is the use of Graph Neural Networks (GNNs) to enhance KG representations, allowing for better integration with dense retrieval methods. This thesis will build upon these trends by focusing on the practical integration of KGs into traditional RAG pipelines and assessing their impact on retrieval accuracy, latency, and explainability.  **Research Content**  This research proposes a novel pipeline to optimize Retrieval-Augmented Generation (RAG) systems by combining hybrid retrieval methods, graph-based community detection, and re-ranking mechanisms. The system is designed to address critical limitations in traditional RAG approaches, such as sub-optimal retrieval relevance, poor scalability, and high latency. By integrating these enhancements, the proposed system can better retrieve and organize relevant information for downstream generative tasks, making it a robust and efficient solution for knowledge-intensive applications.  Traditional RAG systems rely heavily on dense retrieval methods, which use embeddings to search for semantically similar content in vector spaces. However, such methods struggle with handling exact matches, rare terms, and out-of-vocabulary entities. Sparse retrieval methods like BM25, while effective for these cases, fail to capture deep semantic relationships. The hybrid retrieval technique proposed here combines the strengths of dense and sparse methods, ensuring both semantic understanding and precise keyword matching. Furthermore, the system introduces graph-based community detection to group entities and their relationships into structured knowledge representations. This enhances the retrieval process by capturing contextual relevance and interconnections between entities.  The advantages of the proposed system include increased retrieval accuracy, reduced latency, and scalability to large datasets. By using re-ranking techniques and graph-based summarization, the system ensures high-quality retrieval results and context-aware responses. Compared to traditional RAG systems, this approach not only improves retrieval performance but also introduces modularity, enabling easy adaptation to various domains. The integration of graph construction and community detection allows the system to organize knowledge efficiently, reducing redundancy and improving interpretability.  **Research Methodology**  This research aims to address the limitations of traditional Retrieval-Augmented Generation (RAG) systems by proposing a novel and optimized pipeline that integrates hybrid retrieval, graph-based community detection, and re-ranking mechanisms. The methodology is designed to tackle challenges such as retrieving contextually relevant information from vast vector spaces, improving response relevance, and maintaining computational efficiency. By combining advanced retrieval techniques with graph-based organization of knowledge, the system ensures a structured and scalable approach to information retrieval and synthesis.  At a high level, the proposed pipeline begins with preprocessing and preparing unstructured text data, breaking it into manageable units for retrieval. Graph-based methods are then employed to structure the extracted knowledge, with communities of related entities and relationships being detected and summarized. A hybrid retrieval mechanism, combining both dense (embedding-based) and sparse (keyword-based) methods, ensures that the most relevant chunks of information are retrieved, balancing semantic understanding with precision. These retrieved chunks are further refined using a re-ranking process that prioritizes quality and contextual alignment before being passed to a generative model to synthesize a final response.  The core motivation for this approach lies in overcoming the inherent trade-offs between relevance, efficiency, and scalability that traditional RAG systems face. By integrating graph-based knowledge representation and retrieval, this system can organize and filter information more effectively, leading to better quality outputs with reduced computational overhead. The methodology also emphasizes modularity, allowing individual components, such as graph construction or retrieval models, to be easily adapted or replaced based on specific domain requirements.  The following sections describe each stage of the pipeline in detail, outlining the technical processes, formulas, and tools involved in implementing this system.   1. **Data Preparation**   The first step involves preparing textual data for retrieval. Documents are divided into smaller chunks to improve retrieval granularity. Each chunk is capped at a predefined token limit, such as 200-300 tokens:  where is a document, and max\_tokens is the maximum token length. Named Entity Recognition (NER) models are then applied to extract entities *E* and relations *R* from the chunks. Each relation is associated with a confidence score, representing the strength or reliability of the relationship:  These entity-relation pairs form the foundation for constructing a graph representation of the data.   1. **Entity/Relation Confidence Ranking**   Once the entities and relations are extracted, a re-ranking process is performed to filter and prioritize high-confidence relationships. Each entity-relation pair is scored using the NER model, and only pairs with confidence scores exceeding a predefined threshold ( = 0.7 ) are retained:  This ensures that only high-quality relationships are passed to subsequent stages. The ranked and filtered entity-relation pairs are then summarized into instance summaries, which serve as concise representations of the most important entities and their relationships.   1. **Graph Construction and Community Detection**   The next step involves constructing a graph *G* using the ranked entities and relations. The graph is defined as  , where nodes *E* represent entities and edges *R* represent relationships between entities. Each edge is weighted by the confidence score of the corresponding relation:  The constructed graph is then partitioned into communities using the Louvain clustering algorithm, which optimizes the graph’s modularity Q. The modularity is calculated as:  where *m* is the total number of edges, and are the degrees of nodes *i* and *j*, and ensures nodes *i* and *j* are in the same cluster.  To maintain efficiency and relevance, each community is constrained to a maximum of 20 nodes, and only relationships with  > 0.7  are retained.   1. **Community Summarization**   Each detected community is summarized to provide a concise representation of its content. The main entity of the community is identified using degree centrality, which measures the number of direct connections an entity has within the community:  The entity with the highest degree is selected as the main entity:  Using the main entity, the most significant supporting entities, and the strongest relationships, a summary is generated for each community. Summaries are created using a summarization model (e.g., BERT) and adhere to content constraints, ensuring they are concise, relevant, and representative of the community.   1. **Hybrid Retrieval**   To enhance retrieval, a hybrid search combines dense and sparse methods. Dense retrieval uses embeddings    generated by models like Sentence-BERT or text-embedding-ada-002, while sparse retrieval uses BM25 for keyword matching. Given a query  q , dense similarity score and sparse retrieval score are calculated as:  The final hybrid retrieval score    is computed as a weighted combnation of and :  whereis a tunable parameter controlling the balance between dense and sparse retrieval. This hybrid score ensures the best balance between semantic and keyword-based relevance.   1. **Query-Graph Similarity Ranking**   The retrieved communities are further re-ranked based on their similarity to the query. This process uses a similarity function to evaluate the relevance of each community to the query *q*:  The communities with the highest similarity scores are prioritized for subsequent stages.   1. **Subgraph Retrieval**   The top-k ranked subgraphs are retrieved, along with their corresponding community summaries. These subgraphs form the basis of the hypothesis for response generation. The hypothesis combines the most relevant information from the top-k subgraphs and their summaries to create a coherent input for the generative model:  **8. Response Generation**  The final stage synthesizes a response by combining top-k retrieved chunks and community summaries. The generative model (e.g., Qwen or Llama) takes these inputs to produce the output:  where  *H*  is a concatenated representation of retrieved chunks and summaries. The generated response is post-processed for fluency and coherence, ensuring high-quality answers.  **9. Evaluation**  The system’s performance is evaluated across three dimensions:   * Retrieval Accuracy: Metrics like Mean Average Precision (MAP) and Recall@k are used:   where  *Q* is the set of queries,    is the top-k retrieved results, and  *G* is the ground truth.   * Efficiency: Latency and memory usage are recorded to evaluate system efficiency. * Generation Quality: BLEU and ROUGE scores, along with human evaluation, measure the fluency and relevance of generated responses.   **Technical Roadmap**  1. Tools and Frameworks: The research utilizes open-source frameworks like LlamaIndex, LangChain, and PyTorch Geometric for implementing hybrid search and graph-based retrieval. Prebuilt KGs like ConceptNet and DBpedia will be used for initial experiments, with options to construct custom KGs as needed.  2. Pipeline Integration: LlamaIndex and LangChain will be employed to build and optimize the KG-enhanced RAG pipeline, ensuring scalability and efficiency.  3. Evaluation Metrics: The system will be evaluated on metrics such as retrieval accuracy, relevance, latency, and explainability to measure its performance against traditional RAG systems.  **Feasibility Analysis**  The research is feasible within the given timeframe and resources. The use of open-source tools and prebuilt KGs ensures cost-effectiveness. Computational requirements will be met using Google Colab or equivalent platforms. The availability of benchmark datasets, such as Amazon product reviews, and Natural Question, provides a robust foundation for experimentation. The proposed methods and tools align well with the research objectives, ensuring successful completion.  **Project Features and Innovations**  This research proposes several key features and innovations in the optimization of RAG systems. One of the main features is the integration of KGs into the retrieval pipeline, allowing for enhanced semantic understanding through entity linking and graph-based reasoning. This approach bridges the gap between unstructured text data and structured knowledge, enabling more accurate and context-aware retrieval. Another feature is the hybrid search strategy that combines the strengths of dense vector-based retrieval and KG-based semantic search, optimizing both accuracy and latency.  The research also introduces innovative methodologies to improve the explainability and domain adaptability of RAG systems. By leveraging KGs, the proposed system can provide interpretable insights into how retrieval results are generated, making it more suitable for critical applications such as healthcare and finance. Additionally, this research contributes to the growing field of hybrid search by presenting a novel evaluation framework that systematically measures the impact of KGs on retrieval and generation performance. These innovations aim to advance both the academic understanding and practical applications of RAG systems.  **Progress plan**  The Progress plan for the thesis is divided into four phases. The first month will involve reviewing the literature, preparing the dataset, and constructing or selecting a suitable Knowledge Graph. The next two months will focus on designing and implementing the KG-enhanced RAG pipeline. The fourth and fifth months will be dedicated to conducting experiments and analyzing the results. The final month will be used to compile the findings into the thesis report and prepare for the defense.   |  |  | | --- | --- | | Timeframe | Milestones | | Month 1 | Literature review, dataset preparation, KG construction | | Months 2-3 | KG-enhanced RAG system design and implementation | | Months 4-5 | Experimentation, evaluation, and system optimization | | Month 6 | Final report preparation and thesis defense |   **References**  1. S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A Survey on Knowledge Graphs: Representation, Acquisition, and Applications,” IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 2, pp. 494–514, Feb. 2022. DOI: 10.1109/TNNLS.2021.3070843.  2. P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Proc. 33rd Int. Conf. Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 2020, pp. 9459–9474.  3. W. Xie, X. Liang, Y. Liu, K. Ni, H. Cheng, and Z. Hu, “WeKnow-RAG: An Adaptive Approach for Retrieval-Augmented Generation Integrating Web Search and Knowledge Graphs,” in Proc. KDD Cup CRAG Workshop ’24, Barcelona, Spain, Aug. 2024, pp. 1–8. [Online]. Available: https://arxiv.org/pdf/2401.15884  4. 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,” Preprint. [Online]. Available: https://arxiv.org/html/2404.16130v1  5. T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” arXiv preprint, arXiv:1609.02907, Sep. 2016. [Online]. Available: https://arxiv.org/abs/1609.02907.  6. Z. Wang, Y. Zhang, J. Zhu, and C. Tang, “Knowledge Graph-Augmented Semantic Retrieval for Domain-Specific Tasks,” in Proc. 2023 ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), Taipei, Taiwan, 2023, pp. 2237–2240. DOI: 10.1145/3477495.3532068.  7. Nomic AI, “Embedding Techniques for Hybrid Search,” Nomic Documentation, 2024. [Online]. Available: https://nomic.ai.  8. OpenAI, “Optimizing Hybrid Search in Retrieval-Augmented Generation,” OpenAI Documentation, 2024. [Online]. Available: https://openai.com. | | | |
|  | | | |