KAG: Boosting LLMs in Professional Domains via Knowledge Augmented Generation

Lei Liang[*,1], Mengshu Sun[*,1], Zhengke Gui[*,1], Zhongshu Zhu[1], Ling Zhong[1], Peilong Zhao[1], Zhouyu Jiang[1], Yuan Qu[1], Zhongpu Bo[1], Jin Yang[1], Huaidong Xiong[1], Lin Yuan[1], Jun Xu[1], Zaoyang Wang[1], Zhiqiang Zhang[1], Wen Zhang[2], Huajun Chen[2], Wenguang Chen[1], Jun Zhou[†,1]

{leywar.liang, mengshu.sms, zhengke.gzk, jun.zhoujun}@antgroup.com

[1]Ant Group Knowledge Graph Team, [2]Zhejiang University

Github:**https://github.com/OpenSPG/KAG**

Abstract

The recently developed retrieval-augmented generation (RAG) technology has enabled the efficient construction of domain-specific applications. However, it also has limitations, including the gap between vector similarity and the relevance of knowledge reasoning, as well as insensitivity to knowledge logic, such as numerical values, temporal relations, expert rules, and others, which hinder the effectiveness of professional knowledge services. In this work, we introduce a professional domain knowledge service framework called Knowledge Augmented Generation (**KAG**). KAG is designed to address the aforementioned challenges with the motivation of making full use of the advantages of knowledge graph(KG) and vector retrieval, and to improve generation and reasoning performance by bidirectionally enhancing large language models (LLMs) and KGs through five key aspects: (1) LLM-friendly knowledge representation, (2) mutual-indexing between knowledge graphs and original chunks, (3) logical-form-guided hybrid reasoning engine, (4) knowledge alignment with semantic reasoning, and (5) model capability enhancement for KAG. We compared KAG with existing RAG methods in multihop question answering and found that it significantly outperforms state-of-the-art methods, achieving a relative improvement of 19.6% on hotpotQA and 33.5% on 2wiki in terms of F1 score. We have successfully applied KAG to two professional knowledge Q&A tasks of Ant Group, including E-Government Q&A and E-Health Q&A, achieving significant improvement in professionalism compared to RAG methods. Furthermore, we will soon natively support KAG on the open-source KG engine OpenSPG, allowing developers to more easily build rigorous knowledge decision-making or convenient information retrieval services. This will facilitate the localized development of KAG, enabling developers to build domain knowledge services with higher accuracy and efficiency.

1 Introduction

Recently, the rapidly advancing Retrieval-Augmented Generation (RAG)[1, 2, 3, 4, 5] technology has been instrumental in equipping Large Language Models (LLMs) with the capability to acquire

[1], *: These authors contributed equally to this work.

[2], †: Corresponding author.

domain-specific knowledge. This is achieved by leveraging external retrieval systems, thereby significantly reducing the occurrence of answer hallucinations and allows for the efficient construction of applications in specific domains. In order to enhance the performance of the RAG system in multi-hop and cross-paragraph tasks, knowledge graph, renowned for strong reasoning capabilities, have been introduced into the RAG technical framework, including GraphRAG[6], DALK[7], SUGRE[8], ToG 2.0[9], GRAG[10], GNN-RAG [11] and HippoRAG[12].

Although RAG and its optimization have solved most of the hallucination problems caused by a lack of domain-specific knowledge and real-time updated information, the generated text still lacks coherence and logic, rendering it incapable of producing correct and valuable answers, particularly in specialized domains such as law, medicine, and science where analytical reasoning is crucial. This shortcoming can be attributed to three primary reasons. Firstly, real-world business processes typically necessitate inferential reasoning based on the specific relationships between pieces of knowledge to gather information pertinent to answering a question. RAG, however, commonly relies on the similarity of text or vectors for retrieving reference information, which may lead to incomplete and repeated search results. secondly, real-world processes often involve logical or numerical reasoning, such as determining whether a set of data increases or decreases in a time series, and the next token prediction mechanism used by language models is still somewhat weak in handling such problems.

In contrast, the technical methodologies of knowledge graphs can be employed to address these issues. Firstly, KG organize information using explicit semantics; the fundamental knowledge units are SPO triples, comprising entities and the relationships between them[13]. Entities possess clear entity types, as well as relationships. Entities with the same meaning but expressed differently can be unified through entity normalization, thereby reducing redundancy and enhancing the interconnectedness of knowledge [14]. During retrieval, the use of query syntax (such as SPARQL[15] and SQL[16]) enables the explicit specification of entity types, mitigating noisy from same named or similar entities, and allows for inferential knowledge retrieval by specifying relationships based on query requirements, as opposed to aimlessly expanding into similar yet crucial neighboring content. Meanwhile, since the query results from knowledge graphs have explicit semantics, they can be used as variables with specific meanings. This enables further utilization of the LLM's planning and function calling capabilities [17], where the retrieval results are substituted as variables into function parameters to complete deterministic inferences such as numerical computations and set operations.

To address the above challenges and meet the requirements of professional domain knowledge services, we propose **Knowledge Augmented Generation(KAG)**, which fully leverages the complementary characteristics of KG and RAG techniques. More than merely integrating graph structures into the knowledge base process, it incorporates the semantic types and relationships of knowledge graph and the commonly used Logical Forms from KGQA (Knowledge Graph Question Answering) into the retrieval and generation process. As shown in Figure 1, this framework involves the optimization of the following five modules:

- **We proposed a LLM friendly knowledge representation framework LLMFriSPG**. We refer to the hierarchical structure of data, information, and knowledge of DIKW to upgrade SPG to be friendly to LLMs, named LLMFriSPG, to make it compatible with schema-free information extraction and schema-constrained expert knowledge construction on the same knowledge type (such as entity type, event type), and supports the mutual-indexing representation between graph structure and original text chunks, which facilitates the construction of graph-structure-based inverted index and facilitates the unified representation, reasoning, and retrieval of logical form.
- **We proposed a logical-form-guided hybrid solving and reasoning engine**. It includes three types of operators: *planning, reasoning* and *retrieval*, transforming natural language questions into a problem-solving process that combines language and symbols. Each step in the process can utilize different operators such as exact match retrieval, text retrieval, numerical computation, or semantic reasoning, thereby achieving the integration of four distinct problem-solving processes: retrieval, KG reasoning, language reasoning, and numerical computation.
- **We proposed a knowledge alignment approach based on semantic reasoning**. Define domain knowledge as various semantic relations such as *synonyms, hypernyms*, and *inclusions*. Semantic reasoning is performed in both offline KG indexing and online retrieval