Application of search enhancement generation to large prediction models: a literature review

In this literature review, we comprehensively examine the evolution and framework of the RAG model, delve into its core components, and provide a thorough evaluation of the model. We then conduct a comparative analysis with other relevant studies. Finally, we systematically outline the challenges that RAG may encounter in the future along with potential directions for development.

# 1.Overall introduction of THE RAG model

RAG is an important language generation technology aimed at addressing some of the challenges faced by large language models (LLMs), such as hallucination, knowledge obsolescence, and opaque and untraceable reasoning processes. It enhances the accuracy and credibility of generated content by retrieving knowledge from external databases, making it particularly suitable for knowledge-intensive tasks and allowing for continuous knowledge updates and integration of domain-specific information

# 2.The development process and paradigm of the RAG model

## Development trajectory

The early development of RAG was associated with the rise of Transformer architectures, initially focusing on enhancing language models through pre-trained models (PTM) combined with additional knowledge, emphasizing improvements in pre-training techniques. With the emergence of ChatGPT, LLMs demonstrated powerful contextual learning capabilities, leading RAG research to shift towards providing better information to LLMs during inference stages to address more complex and knowledge-intensive tasks. Later, enhancements to RAG were no longer limited to the inference stage but began to be more closely integrated with LLM fine-tuning techniques.

## Research paradigms

Naive RAG: This is the earliest method, following the traditional indexing, retrieval, and generation process, also known as the "Retrieve-Read" framework. The indexing stage cleans and extracts data of different formats, converting them into a unified plain text format, then divides them into blocks and encodes them as vectors stored in vector databases. The retrieval stage calculates similarity scores based on user queries, obtaining the top K most similar blocks. The generation stage synthesizes the query with selected documents into a coherent prompt, allowing large language models to generate answers. This paradigm suffers from issues such as low retrieval accuracy and recall rates, potential hallucinations during generation, and difficulties in integrating retrieval information.

Advanced RAG: To overcome the limitations of Naive RAG, the focus is on improving search quality by adopting pre-search and post-search strategies. The pre-search process optimizes index structure and primary queries, such as enhancing data granularity, optimizing index structure, and adding metadata; the post-search process includes reordering blocks and compressing contexts to better integrate retrieved information.

Modular RAG: This architecture is more adaptable and versatile introducing multiple strategies and new modules. The new modules include search module, RAG-Fusion, memory module, routing module, prediction module, and task adapter module, which enhance retrieval and processing capabilities. At the same time, it features new patterns that allow module replacement or reconfiguration, such as Rewrite-Retrieve-Read, Generate-Read, Recite-Read patterns, and hybrid retrieval strategies.

# Core components of the RAG model

1. Retrieval

Retrieval Source: including text, semi-structured data, and structured data, as well as research that utilizes content generated by LLMs for retrieval. The granularity of retrieval ranges from fine to coarse, including Token, Phrase, Sentence, Proposition, Chunks, Document, etc., and on knowledge graphs also includes Entity, Triplet, sub-graphs, etc., and the granularity of retrieval can be adjusted according to downstream tasks.

Index Optimization (Indexing Optimization): Common partitioning strategies involve dividing documents into fixed numbers of segments, but this approach balances semantic integrity and context length. Methods such as Small2Big have also been proposed. Additionally, metadata information (such as page numbers, file names, authors, etc.) can be added to enrich the segments, which can then be used for retrieval filtering or manually constructed. Query Optimization (Query Optimization): This includes query expansion (such as multi-query expansion using LLMs,subquery planning, Chain-of-Verification validation for extended queries), query transformation (such as rewriting queries using LLMs,generating new queries based on original queries using LLMs), and query routing (such as routing based on metadata or semantic information). Furthermore, indexing with hierarchical structures (such as hierarchical index structures and knowledge graph indexes) can enhance information retrieval.

Embedding: Retrieval is achieved through calculating the similarity between query and document blocks, embedding models include sparse encoders (such as BM25) and dense retrievers (such as pre-trained language models based on BERT architecture), as well as some new prominent embedding models. Hybrid retrieval methods can be adopted, and in certain cases, the embedding model may need to be fine-tuned.

Adapter: Some methods introduce external adapters to assist alignment, such as the lightweight prompt retriever trained by UPRISE, the general adapter of AAR, and the plug-and-play reward-driven context adapter of PRCA.

1. Generation (Generation)

Context Curation: includes reordering (reordering document blocks using rule-based methods or model-based methods) and context selection/compression (such as detecting and removing unimportant tags using small language models, or training information extractors, information condensers, etc.).

LLM Micro-Tuning (LLM Fine-tuning): Fine-tuning LLMs according to scene and data characteristics can yield better results such as providing additional knowledge in specific domains adjusting the models input and output aligning LLM outputs with human or retriever preferences through reinforcement learning and coordinating LLM micro-tuning with retriever micro-tuning.

1. Enhancement process (Augmentation Process)

Iterative Retrieval: Based on the initial query and the generated text, it repeatedly searches the knowledge base to provide a more comprehensive knowledge base for LLM, but may be affected by semantic discontinuity and irrelevant information accumulation.

Recursive Retrieval: It improves the depth and relevance of search results by iteratively refining the search query. It can be used in combination with multi-hop retrieval technology and is useful in complex search scenarios.

Adaptive retrieval (Adaptive Retrieval): such as Flare and Self-RAG, enables LLM to actively determine the best time and content of retrieval, improving the efficiency and relevance of information acquisition.

# 4.Tasks and evaluation of the RAG model

For downstream Task,the core task is question answering (QA), including traditional single jump/multi jump QA, multiple selection, specific domain QA and long text scene QA, as well as extended to information extraction, dialogue generation, code search and other downstream tasks.

And the evaluation Target,this includes retrieval quality (measured by standard indicators of search engines, recommendation systems and information retrieval systems) and generation quality (assessed according to the fidelity, relevance, harmlessness and accuracy of generated answers).

What about evaluation Aspects,three main quality scores (context relevance, answer fidelity, and answer relevance) and four basic capabilities (noise robustness, negative rejection, information integration, and counterfactual robustness) are emphasized.

And the lastly is evaluation Benchmarks and Tools,there are a series of benchmarks and tools, such as RGB, RECALL, CRUD benchmarks, and RAGAS, ARES, TruLens tools, for evaluating the RAG model.

# 5.Comparison between THE RAG model and other related studies

## Compared to fine tuning

RAG is analogous to providing a customized textbook for model information retrieval suitable for precise information retrieval tasks with the advantages of real-time knowledge updates, effective utilization of external knowledge sources, and high interpretability but it has higher latency and ethical issues in data retrieval. Fine-tuning is similar to students learning over time suitable for scenarios that require replicating specific structures, styles, or formats, being more static, requiring retraining updates, but capable of deeply customizing model behavior and style. In multiple evaluations of different knowledge-intensive tasks, RAG typically outperforms unsupervised fine-tuning in both existing and new knowledge areas, and the two can complement each other.

## Compared with early RAG related studies

The early proposed RAG model consists of two parts: the retriever and the generator. The retriever adopts a dual-encoder architecture based on DPR, while the generator is based on BART-large. Two models, RAG-Sequence and RAG-Token, were proposed, each with distinct characteristics in training and decoding methods. These models have achieved good results in tasks such as open-domain question answering, abstract question answering, Jeopardy question generation, and fact verification, indicating that combining parametric and non-parametric memory is beneficial for knowledge-intensive tasks, and that non-parametric memory can be updated, enabling the model to adapt to changes in world knowledge.

# 6.Challenges faced by the RAG model and future development directions

There remain significant challenges in comparing Retrieval-Augmented Generation to long text processing and competition in long document Question and Answer systems. As the context window of Large Language Models 、continues to expand, long document Q&A may increasingly operate independently of RAG, as LLMs can now process documents with over 200,000 tokens directly. However, noise and contradictory information during the retrieval process pose a threat to the output quality of RAG. Research has shown that the inclusion of irrelevant documents can sometimes unexpectedly improve accuracy, which contradicts intuition and underscores the need to enhance RAG's robustness against such information. Future optimization of RAG could take a hybrid approach, incorporating fine-tuning strategies. This includes determining the optimal integration methods (sequential, alternating, or end-to-end joint training) and leveraging the strengths of both RAG and fine-tuned models. Additionally, introducing specialized small models and fine-tuning them based on RAG system outcomes represents another promising trend. Moreover, developing new RAG methodologies for ultra-long contexts is essential to address more complex and comprehensive problems.