2018 words | 6 minutes

Retrieval Enhancement Generation RAG Introduction

1. What is RAG

Large language models (LLMs) are more powerful than traditional language models, but in some cases they may still not provide accurate answers. In order to address a series of challenges faced by large language models when generating text and improve the performance and output quality of the models, researchers proposed a new model architecture: **Retrieval-Augmented Generation (RAG)** . This architecture cleverly **integrates relevant information retrieved from a huge knowledge base, and based on this, guides large language models to generate more accurate answers** , thereby significantly improving the accuracy and depth of the answers.

The main problems currently faced by LLM are:

- **Information bias/illusion:** LLM sometimes generates information that is inconsistent with objective facts, resulting in inaccurate information received by users. RAG assists the model generation process by retrieving data sources to ensure the accuracy and credibility of the output content and reduce information bias.

- **Knowledge update lag:** LLM is trained based on static data sets, which may cause the model's knowledge update to lag and fail to reflect the latest information dynamics in a timely manner. RAG retrieves the latest data in real time to maintain the timeliness of the content and ensure the continuous updating and accuracy of information.

- **Content is not traceable:** LLM-generated content often lacks a clear source of information, which affects the credibility of the content. RAG links the generated content with the retrieved original materials, enhancing the traceability of the content and thus increasing users' trust in the generated content.

- **Lack of domain expertise:** LLM may not be very effective in handling domain-specific expertise, which may affect the quality of its answers in related fields. RAG retrieves relevant documents in a specific field and provides the model with rich contextual information, thereby improving the quality and depth of answering questions in the domain of expertise.

- **Reasoning ability limitations:** When faced with complex questions, LLM may lack the necessary reasoning ability, which affects its understanding and answering of the questions. RAG combines the retrieved information with the generative ability of the model, and enhances the model's reasoning and understanding ability by providing additional background knowledge and data support.

- **Limited adaptability to application scenarios:** LLM needs to be efficient and accurate in a variety of application scenarios, but a single model may not be able to fully adapt to all scenarios. RAG enables LLM to flexibly adapt to various application scenarios such as question-answering systems and recommendation systems by retrieving data for the corresponding application scenarios.

- **Weak ability to process long texts:** LLM is limited to a limited context window when understanding and generating long content, and must process content sequentially. The longer the input, the slower the speed. RAG strengthens the model's understanding and generation of long contexts by retrieving and integrating long text information, effectively breaking through the limitation of input length, while reducing call costs and improving overall processing efficiency.

2. RAG Workflow

RAG is a complete system whose workflow can be simply divided into four stages: data processing, retrieval, enhancement, and generation:

1. **Data processing stage**
    1. Clean and process the raw data.
    2. Convert the processed data into a format that can be used by the retrieval model.
    3. The processed data is stored in the corresponding database.
2. **Retrieval stage**
    1. The user's question is input into the retrieval system and relevant information is retrieved from the database.
3. **Enhancement Phase**
    1. The retrieved information is processed and enhanced so that it can be better understood and used by the generative model.
4. **Generation phase**
    1. The augmented information is fed into the generative model, which generates answers based on the information.

3. RAG VS Finetune

RAG and fine-tuning (Finetune) are two mainstream methods for improving the performance of large language models.

Fine-tuning : Further training a large language model on a specific dataset to improve the model's performance on a specific task.

The comparison between RAG and fine-tuning can be found in the following table (source: [1][2])

Feature Comparison	RAG	Fine-tuning
Knowledge Update	Directly update the retrieval knowledge base without retraining. The cost of information update is low and suitable for dynamically changing data.	Retraining is usually required to keep knowledge and data updated. The update cost is high and it is suitable for static data.
External knowledge	Skilled in leveraging external resources, particularly suited to working with documents or other structured/unstructured databases.	Learn external knowledge into LLM.
data processing	The requirements for data processing and operation are extremely low.	Relying on building a high-quality dataset, a limited dataset may not significantly improve the performance.
Model customization	Focuses on information retrieval and incorporating external knowledge, but may not adequately customize model behavior or writing style.	LLM behavior, writing style, or specific domain knowledge can be tailored to a particular style or terminology.
Explainability	It can be traced back to the specific data source and has good explainability and traceability.	Black box, relatively low interpretability.
Computing resources	Additional resources are required to support the search mechanism and maintenance of the database.	Relying on high-quality training datasets and fine-tuning targets, it places high demands on computing resources.
Inference Latency	Increased the time consumption of the retrieval step	Time consumption for simple LLM generation

Feature Comparison	RAG	Fine-tuning
Reduce hallucinations	The answers are generated by retrieving real information, which reduces the probability of hallucination.	Models that learn from domain-specific data can help reduce hallucinations, but they may still hallucinate when faced with unseen inputs.
Ethical Privacy	Retrieving and using external data may raise ethical and privacy issues.	Sensitive information in training data needs to be properly handled to prevent leakage.

IV. RAG’s Success Stories

RAG has achieved success in many fields, including question-answering systems, dialogue systems, document summarization, document generation, etc.

We will introduce the application of RAG in detail in the third part. We will disassemble the existing mature RAG cases and have a deeper understanding of RAG with you.

1. The Datawhale Knowledge Base Assistant is a combination of the content of this course. It is based on ChatWithDatawhale , a Datawhale content learning assistant created by Sanbu , and adjusts the architecture to the LangChain architecture that is easy for beginners to learn. It is an LLM application that encapsulates large model APIs from different sources with reference to the content of Chapter 2. It can help users communicate smoothly with DataWhale's existing warehouses and learning content, thereby helping users quickly find the content they want to learn and the content they can contribute.
2. Tianji is a free, non-commercial artificial intelligence system developed by **SocialAI** . You can use it to perform tasks involving traditional social skills, such as how to toast, how to say nice things, how to be good at dealing with people, etc., to improve your emotional intelligence and core competitiveness. We firmly believe that only social skills are the core technology of future AI, and only AI that is good at dealing with people has the opportunity to move towards AGI. Let us work together to witness the advent of general artificial intelligence. —— "Tianji cannot be leaked."

> **In this chapter, we have a brief understanding of RAG. In the next chapter, we will introduce a commonly used RAG development framework LangChain.**

【Reference content】：

1. Retrieval-Augmented Generation for Large Language Models: A Survey
2. Retrieval-augmented generation techniques for large language models: A review

‹ Previous chapter

1. Introduction to Large Language Model (LLM) Theory

Next Chapter ›

3. Introduction to LangChain