How To Build A RAG-Based LLM Application From Scratch and Deploy It In Production

As we learned in the previous lectures, RAG is a powerful and flexible method for retrieving and generating natural language with large language models. However, RAG is not just a theoretical concept or a research paper. RAG can also be used to build real-world applications that can solve various problems and provide value to users. For example, you can use RAG to build a chatbot that can answer questions, a summarizer that can condense information, a recommender that can suggest products, a writer that can generate content, and many more.

But how do you build a RAG-based LLM application from scratch and deploy it in production? What are the steps and the tools that you need to follow and use? What are the best practices and the challenges that you need to consider and overcome? These are some of the questions that we will answer in this lecture.

To build a RAG-based LLM application from scratch and deploy it in production, we need to follow some steps. These steps are:

- Step 1: Define the problem and the solution. We need to decide what kind of problem we want to solve with the RAG-based LLM application, and what kind of solution we want to provide to the users. For example, we may want to solve the problem of finding relevant information from a large and complex knowledge source, and provide a solution that can generate concise and accurate answers or summaries. We also need to define the scope and the requirements of the problem and the solution, such as the target audience, the use cases, the features, the metrics, etc.
- Step 2: Design the architecture and the interface. We need to decide how to structure and organize the RAG-based LLM application, and how to interact with the users. For example, we may want to design a web-based application that consists of a front-end, a back-end, and a database, and that allows the users to enter queries and receive responses through a graphical user interface. We also need to decide how to integrate the RAG framework and its components, such as the query encoder, the retriever, the generator, and the knowledge source, into the architecture and the interface.
- Step 3: Implement the code and the logic. We need to write the code and the logic that implements the functionality and the behavior of the RAG-based LLM application. For example, we may want to use Python as the programming language, and use frameworks and libraries such as Hugging Face Transformers, Ray, PyTorch, Flask, etc. to implement the code and the logic. We also need to use the best coding practices and standards, such as documentation, testing, debugging, etc. to ensure the quality and the reliability of the code and the logic.
- Step 4: Test and debug the application. We need to test and debug the RAG-based LLM application to ensure that it works as expected and meets the requirements. For example, we may want to use tools and methods such as unit testing, integration testing, system testing, user testing, etc. to test and debug the application. We also need to identify and fix any errors, bugs, or issues that may occur during the testing and debugging process.

- Step 5: Deploy and monitor the application. We need to deploy and monitor the RAG-based LLM application to make it available and accessible to the users. For example, we may want to use tools and services such as Docker, Kubernetes, AWS, Anyscale, etc. to deploy and monitor the application. We also need to ensure the scalability, availability, security, and performance of the application, and handle any challenges or problems that may arise during the deployment and monitoring process.
- Step 6: Iterate and improve the application. We need to iterate and improve the RAG-based LLM application to meet the changing needs and expectations of the users. For example, we may want to collect and analyze the feedback and the data from the users, and use them to improve the features, the functionality, the quality, and the performance of the application. We also need to keep up with the latest developments and trends in the field of RAG and LLMs, and incorporate them into the application.