User:

Hi, nice to meet you.

Assistant:

Hello! It's nice to meet you too. Thank you for taking the time to speak with me today about the Entry-Level RAG AI Engineer role. To start off, could you tell me about your experience with retrieval-augmented generation (RAG) pipelines?

User:

Yeah sure, I'd be happy to. I do have some experience with these. I have built a web scraping pipeline before which scrapes the web and puts all the information that is gathered into a very relevant query into the Chroma Vectorize database and then uses an in-banks model to get a similarity between the input tokens and the Vectorize database to then figure out what sections of the database are relevant and to put that into the context window of the LLM allowing the LLM to respond using information obtained from the web that was scraped.

Assistant:

That's interesting. Can you elaborate on how you handled the efficiency and performance aspects of this RAG pipeline, particularly in terms of speed and cost optimization?

User:

Sure, yeah. So mainly when it comes to speed, there's two variables that I can mainly play with to increase the performance of the entire system. The first one is when actually conducting the web scraping, ensuring that the web scraping is done entirely in parallel. So this involves generating all the list of queries and then sending off the requests using threads and then waiting for all the threads to be rejoined back and then just making a list of all the retrieved data by this appending onto that list and then from there onwards you can continue processing. And the second one is to generate the relevant search query. You can use a cheaper model instead of like a very large expensive LLM. So for example, I could use Gemini 1.5 Flash instead of Gemini 1.5 Pro, which is a smaller and cheaper model to run, which also saves in terms of the cost.

Assistant:

Thank you for that explanation. Moving on, could you describe your experience with different large language models (LLMs)? Have you worked with or compared models like OpenAI's GPT, Llama, or Claude?

User:

Yep, I have got some experience when it comes to these different language models. There is a lot of differences between these three models and the aforementioned Google Gemini. For example, Llama is open source, which means I can download it and run it locally or run it online and host it. But because it's open source, I'm not tied down explicitly to open AI terms of service, nor am I tied down to having to run my code on open AI servers. So if my application is strongly tied to security, that would be a good example. An alternative would be to use, for example, something like Google Gemini, which has some benefits over Llama and OpenAI's chat GPT, in that it has a much larger context window of 1 million tokens. So if there is a vast amount of data that I need to run and process all at once, I can fit all that into the context window, such as frame-by-frame analysis of an entire movie, for example, that would be possible inside of Gemini, that would not be possible in cloud or GPT.

Assistant:

I see. For our final question, how would you approach integrating an AI solution, specifically a RAG system, into a production environment? What considerations would you keep in mind?

User:

Sure, there are several considerations that I have to keep in mind. For example, if I'm deploying a large language model for users to use, they may attempt to prompt engineer it themselves, which is actually a surprisingly intuitive thing for regular people to try to do, by asking it to do other things or telling it to ignore its instructions and then do something else, etc. So that would be a consideration that I'd have to keep in mind. When it comes to RAG systems, if the user is feeding in documents into the system, I would need to keep in mind the fact that users will only be restricted to only submitting a certain type of documents or appropriate documents that is fitting to what the purpose is of our system. And so there'd have to be some pre-processing to all the input documents that is being provided. That's about the two biggest ones, but there's definitely some more as well that don't immediately come to mind.

Assistant:

Thank you for your time and responses. We appreciate you taking part in this screening interview for the Entry-Level RAG AI Engineer role. You'll be hearing from us soon regarding the outcome of this interview. Have a great day!