User:

Hello.

Assistant:

Hello! Thank you for taking the time to speak with me today about the Entry-Level RAG AI Engineer role. I'm looking forward to learning more about your experience and skills. To start off, could you tell me about any experience you have with setting up and optimizing retrieval-augmented generation (RAG) pipelines?

User:

I'm sure. So I'm currently working on my final year dissertation which is to build a LLM screening system that will screen candidates and assess whether they are fit to move on to the next round. So to achieve this one of the steps involved checking the accuracy of the candidates answer. To check the accuracy of the answer what I basically did was set up a rag pipeline. That pipeline involved getting each, getting the transcript of each question and answer and then dividing the answer into searchable queries using an LLM, using those searchable queries to web script Google links and put it in the context window of a vector store. Now I created another rag chain which analyzes the accuracy of each answer keeping the training data of the LLM that was already there and the web script data that was in the vector store that it searches the most relevant information from the vector store and doesn't get fetch everything from the vector store and puts it in the context of the LLM and then it judges the accuracy of the answer. Now that we have the accuracy of the answer and we can assess the how right or wrong the candidates per answer. This is just one part of the sentiment analysis. I can go on with the entire sentiment analysis if you want later on.

Assistant:

Thank you for sharing that experience. It's certainly relevant to the role. Could you elaborate on your experience with different large language models? Have you worked with or compared various LLMs like OpenAI, Llama, or Claude in your projects?

User:

Yes, sure. So in my current project with the Screening LLM, I'm currently using Claude and OpenAI. What I found is Claude tends to be more accurate and comes to a more descriptive answer with a single prompt rather than OpenAI. An example would be, so if I tell Claude to write a bare bones frontend for a UI, Claude can do it in a single go, whereas OpenAI struggles to do it in a single go and sometimes have to give multiple prompts. So Claude is better at understanding code, better than OpenAI. However, OpenAI does have a mini model called GPT-4.0 mini that uses less, that I think uses less parameters than the normal GPT-4.0, so it's quicker. So I can also some of the tasks to GPT-4.0 when I require real-time analysis. An example of this would be if I had required, if I required that each answer from the candidate would have to be processed in real time rather than post the interview, I would use GPT-4.0 in that case.

Assistant:

That's an interesting comparison of Claude and OpenAI models. You've clearly gained some practical insights. Now, could you tell me about your experience with AWS services, particularly in a production environment?

User:

Yes, so for AWS, I have previously deployed applications when I was working for AMEX into AWS, and also in private servers. So for AWS, some of the experiences I have in deploying are generating a Docker file that can be uploaded directly to an AWS server. So whenever we require anything for the project, it can be downloaded specifically from the company's repository in the AWS server. Also, we can deploy entire applications into AWS servers, and depending on the load balancing settings, it can be upscaled as and when traffic to the web application increases or decreases. This is done automatically through configuration parameters. I am yet to learn all the configuration parameters. That is the duty of Debugs, but I am keen to learn more on that topic.

Assistant:

Thank you for sharing your experience with AWS. Lastly, could you describe a challenging problem you've encountered in your AI or machine learning projects and how you approached solving it?

User:

I'm sure so we had a group project for developing Sorry, we had a group project for beating a game called jumping using neat which is neural evolution of augmented topologies So what it does is basically? Start with a few basic generations of neural networks that are randomly generated and it sees the neural networks that perform best for solving the task and those neural networks reproduced among themselves to produce the next generation so basically that the best genes of the best from each of the parents are Move forward in the next generation as well as the top performing network in the first generation and this continues on and on and on until We get the most optimum solution also I forgot to mention that we have a fitness function defined for this Which calculates how good a neural network performs so the better the neural network performs the more Chances of it reaching the next generation or the more chances of it More chances of it Producing with another neural network to move on to the next generation if that makes sense So we had to we had used this to beat the game jumping One of the challenges was to come up with a fitness function that is how the Individual king in the game would perceive the environment around it so it needs to jump to the highest level possible right so one method of the one method of getting the fitness function was to calculate the height that the King that the Kings reached at the end of each level however we had to feed in the Input to the neural network in such a way that it recognized the geometry of the game that was pretty hard to overcome We finally came up with a solution of distance from the The shortest distance from the King that is the shortest diagonal distance the shortest vertical distance the shortest horizontal distance And we also rewarded it for reaching a higher height and also penalized it if It fell down from a certain platform Yes, so that is how we were working the challenges and the After Incorporating the network into incorporating no incorporating is not the term I'm looking for after Feeding the Inputs in the form of the XY coordinates of the platform it was able to significantly Increase the performance of the network by increase the performance. I

mean it was able to Increase the fitness function generation after generation until the King reached the final stage

Assistant:

Thank you for sharing that detailed example. It's clear you have experience with complex AI problems and innovative approaches. Thank you for your time today. We'll be in touch soon regarding the next steps in the interview process.