AI Engineering

Senior ML Engineer Tech Assessment

Introduction

Thank you for accepting the technical challenge for Loka. The goal of this challenge is to evaluate your problem-solving skills, technical knowledge, and ability to put it into practice.

We recommend that you first think through your solution and draft it before implementing as much as possible. Note that Loka works closely with AWS, so we will highly value solutions that are cloud-compatible, particularly with AWS. We do not expect you to spend money or have a powerful machine at your disposal, so feel free to not deploy the most expensive/compute-intensive parts of your architecture and just mock them (locally, on colab, or however makes sense to you).

We provide you with a dataset of documents and a description of a general case that is common in the field and based on an internal project named Clementine. Note that for simplicity, the dataset provided is AWS documentation (same as Clementine) that is widely available.

Remember that the goal of this challenge is for you to shine and show us what you know. You should not concern yourself too much if the output of your system isn't the best you can do as long as you clearly explain your architecture and design process. If you feel you have relevant skills that aren't being showcased in this challenge, feel free to send us an email and explain the changes to the scenario you would propose and the skills you would like to showcase.

Scenario

Company X has a large amount of documentation that their developers need to navigate. They have noticed that their developers often spend significant amounts of time searching through documentation or asking other developers simple questions that are in the documentation. The Company has reached out to Loka to assist with building a tool to address this issue.

After some discussion, it is agreed that the first step of the collaboration should be a POC whose goal would be to prove that the system would significantly shorten the amount of time developers spend looking through documentation. The POC will cover only a subset of their data and will initially just be applied to one of the teams.

Upon further investigation, they mention that their main goal would be to have a system that could assist developers with parts of the documentation they aren't familiar with, as in these cases they typically reach out to coworkers with some pretty simple questions. This has several issues, among them experienced members have their work interrupted often and responses are sometimes based on old information, as documentation is often updated, causing problems down the line. They would also like for the system to be able to point the user to further reading, by pointing them towards the source for the response and towards other documents that may be relevant to what they are currently working on. However, this last request is a nice-to-have and not a mandatory feature. They can accept this being implemented later.

The documentation provided for the POC is public (it's AWS documentation) and as such has no limitations on usage. However, the final system will also handle internal documentation that contains sensitive information that has proprietary (can't be shared or accessed externally) and geographical restrictions (can't leave the US).

They also provided some example questions they would like the system to be able to respond to for the POC:

- What is SageMaker?
- What are all AWS regions where SageMaker is available?
- How to check if an endpoint is KMS encrypted?
- What are SageMaker Geospatial capabilities?

Your task is to design the overall solution for Company X. Determine what parts of that solution should be part of the POC and begin implementing the POC.

Deliverable

The output of this exercise is expected to be a private GitHub repository. Please add the following users to the repository:

- @henriqueribeiro
- @caldasdeoliveira
- @tsfelg
- @ricardommarques
- @bojanilijoski
- @bonnec

Suggestions

Since we want to see you at your best, we'll leave you with some tips and questions that we may ask during the interview. Feel free to use this section to help you prepare for the interview.

1. Does your solution solve the company's pain points? What are they?
2. What is the name of the LLM Pattern you've used in this project? Since names are not yet standardized, feel free to elaborate on the pattern you used.
3. What tools did you use? Why did you select them?
4. What model would you use for this use case? Why?
    (a) What did you use for your embeddings? How does that decision affect the performance of your system?
5. How does your system handle out-of-vocabulary (OOV) terms?
6. Would you need to self-host? Explain your decision.
7. How did you chunk the documents provided? Does this decision have any effect on the performance of the system?
8. What is missing for your solution to be production-ready?
9. Is your system able to handle changing information? What would happen if the documentation is updated?

10. How can you evaluate your system?
    (a) How do you evaluate your information retrieval system?
    (b) What would need to be different between evaluation during development and for production?

As always, if you have any questions regarding the scenario, the interview process, are unsure about what to prioritize or implement, need extra information to make meaningful decisions, or any other point that you think the team can help clarify, feel free to send us an email with your questions, and we'll try to help you as much as possible.