AI Solutions Engineer Case Study for Aleph Alpha

23.04.2025

ALEPH ALPHA

Instruction

Congratulations on making it to the case study round! :)

We appreciate your time and effort so far and are really excited to see what you will come up with during this case study.

During this round, we will present you with a case study to give you the opportunity to showcase your skills. It will be open-ended, and you will have the freedom to tackle the problem as you please.

Good luck!

Deliverables

There are two sets of deliverables and a presentation:

- Code Artifact: Present a solution that goes beyond a simple Python script. For example, this could be in the form of a simple API, a Gradio/Streamlit app, or a Python package. Be creative!
- Documentation: Document your assumptions, approach, and considerations while you work on the problem. Include all the results of your experimentation and the shortcomings of your solution. Give a brief explanation of how you would proceed if you had more time.
- Presentation: You should be prepared to present your code artifact and documentation to us in some form. You will have 20 minutes to present your solution, and we will reserve the remaining time for questions. Please note that you don't need to prepare any extra slides. A walkthrough of the code artifact and documentation are sufficient.

What we are providing

Data: Navigate to the data/folder in the repository to find the data you need for the task. In old_tickets you will find tickets that have been resolved before and in new_tickets.csv you will find tickets that you can use test and evaluate your solution.

Case Background

You are working at Aleph Alpha as an AI Solutions Engineer. You have a client who runs IT helpdesks for different companies. A lot of the tickets the client handles are redundant and might just need a little tweak to solve the problem of the client's customer. For each previously resolved ticket, there is a description of how the problem was solved. The issue is that service agents often don't know if and how a problem has been solved before by someone else. Therefore, the client wants to build a new feature that would assist their agents while working on incoming new tickets, utilizing the information from tickets that have been resolved before. The goal is to use previous tickets to aid the agent in finding a solution for incoming new tickets.
Hint: You don't need to provide a solution to the agent; it is enough to provide a direction that might lead the agent to solving the ticket.

Task

Data: First, we are interested in the steps you would take to understand the problem and propose a solution. Second, use the tools of your choice to build a simple prototype that could assist the agent. We will provide a "database" of previous tickets that have been solved and some tickets that you can evaluate your prototype on. The focus of the task should be more on the approach and considerations you make than the quality of the final output. Please don't spend more than 3-5 hours on this task but rather tell us what more you could have done if you had more time.

Hint: Use your time wisely; an imperfect solution is better than an incomplete solution.

Tools available

- You can use any tool of your choice, but we recommend to use the following two models via huggingfaces free API
- Embedding model: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
- Generation model: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
- Huggingface documentation to get you started: https://huggingface.co/blog/getting-started-with-embeddings https://huggingface.co/docs/huggingface_hub/en/package_reference/inference_client#huggingface_hub.InferenceClient.chat_completion