Agent Fulcrum,

The following message is highly confidential, read it carefully and make sure it does not fall into the hands of the Empire.
As you are well aware, there are three literary sagas that are enjoying great success in the Imperial capital of Coruscant. As a propaganda technique, we at the Rebel Alliance plan to design and launch a publicity campaign based on characters from the sagas to inspire people to join our cause. To help us with the task, our spies have managed to create a system that analyses in real time how many people read a text related to any of the three.
In addition, we will provide you with the following information:

- A list of the most important characters in these sagas.
- A model that, given an article, is able to tell which of the three literary sagas it is talking about.

literary sagas.

Aim of the test

The aim of the test is to design a system that allows our analysts to know which characters are the most interesting for people in a specific period. To do so, you will have to design and implement a system (together with the necessary ingest, transformation and storage architecture) that meets the specifications detailed below:
- The output of the system must be the top N entities that have been read about the most in a given saga, as well as the number of people who read about each of them.
- The system must support as optional filters a start and end timestamp. The format of these timestamps is up to you.
- The components of the system are also up to you, including the way in which the data is exposed (HTTP API, database, query interface, etc.).
- Minimally document how to access the data. In case of exposing the data directly through a database, include at least a couple of example queries.

In summary, the system should support a number N of entities, the label of one of the 3 sagas, and (optionally) a start and end timestamp. The expected output will be any data structure containing the N most read entities in that period, together with the total volume of readers.

Details of the resources

As a starting point for the test, the following resources are available: entities.txt: list of the most important entities/characters of the three sagas. docker-compose.yml: docker-compose which raises the following services:
- classification-service: item classification service. It has a route on its port 5000 that expects a POST request with the following body: . The emitted labels are: "got", "lotr", "hp".
- data-streaming-service: generates a data stream, which is published in the events topic of the kafka container. Each of the events measures how many people read a particular article at the time it is posted.
- zookeeper - kafka

Additionally, a container called listener-service will be raised for the sole purpose of allowing you to browse the data stream. In order to access it, simply enter the container with the following command:

```
docker-compose exec listener-service bash
```

and start the following process:

```
python3 kafka_consumer.py
```

This process will display the data stream in real time.

Instructions

The ultimate goal of the test is to see how you perform when working on this kind of problem. Don't worry if the result is not 100% functional. Try to focus on doing things the way you would do them in a real environment, briefly documenting design decisions, system architecture and next steps.

When submitting the test, remember to include the necessary documentation, instructions for executing the result, and send it together with the code and files of your test all inside a single compressed file.

```
 /predict
{"text": "text to be classified"}
```

Translated with www.DeepL.com/Translator (free version)