LLM - Technical Test

As an ML Software Engineer at InstaDeep, you will have the opportunity to work on complex problems that involve a proper understanding of a problem, along with the necessary set of skills to analyze and solve it.

For this technical test, we would like you to help a researcher update her/his paper based on the latest novelties in the field. You will design an application to do so. We will provide a set of related papers that you can use for this task. The application input will be a set of paragraphs. Your objective is to recommend novel papers that would be useful to the researcher to update the set of input paragraphs.

Data

The datasets are in these GCP bucket:
- Raw PDF scientific papers: link
- PDF files extracted with GROBID API: link

We provide the raw pdf data and the extracted text.

Tasks

- Create a vector database with LLM and documents (pdf or extracted text).
- Given the input (a paper's paragraph), get the top 3 most similar papers in the database.
- For each paper, provide a summary.
- Provide an evaluation pipeline that helps assess the quality and accuracy of the output.

Deliverable

Your deliverable should be divided into the following parts:

- Structure project repository with clear instructions to launch the application and best software engineering practices.
- If you provide a git repository, please launch from your repository git bundle create <YOUR_NAME>.bundle --all and send us the resulting <YOUR_NAME>.bundle file.
- If you host your git repository on GitHub/GitLab, please keep it private.

Technical requirements:

- Python 3.6+
- Easily reproducible on a laptop with 16GB of RAM + 4GB GPU
- We recommend using Llama, Langchain or Streamlit, but we leave you the space to make your own choices as well.

- You can use OpenAI or Bard API if you want to do so.

Evaluation

- You won't be evaluated on the final performance of your engine but rather on the methodology you used to tackle this task, so make sure to explain each step.
- We will pay attention to the code quality, documentation and reproducibility.

Compute

In case you need more compute power than locally available on your computer, the following resources provide interesting amounts of computing power for free:

- Google Colab: access to one GPU or one TPU, time limit of 12 hours (kernels are shut down after 12 hours)
- Kaggle notebooks : access to one GPU (NVIDIA P100), time limit of 6 hours Hope you have fun!

Please feel free to contact us if you have any questions, we'll be happy to help.