Modelización de problemas de la Empresa

Problem proposal

Title: Using LLMs for job searching

Description:

Large Language Models (LLMs) have revolutionized Natural Language Processing with their human-like text generation. This is leading companies to use the potential of LLMs to transform their industries by optimizing processes.

In this project, we are given a huge amount of job offers and we need to find the one that best fits our candidate's interests and abilities (Profile 1 and 2). You are allowed to use open-source or proprietary models based on your preference. If you prefer to use an open-source model, we recommend you to use the following: Mistral-7B-quantized.

Profile 1: A female student with a Bachelor's degree in Mathematics and a strong interest in building end-to-end applications. She is more interested in developing backend software than frontend interfaces.

Profile 2: A male student with a Bachelor's and Master's degree in Computer Science. He is passionate about data and is eager to pursue a career as a data engineer at a cutting-edge tech company.

Development details:

You can download the data from https://www.kaggle.com/datasets/ravindrasinghrana/job-description-dataset.

Then:

1. Filter your data so that job offers do not require years of experience. Moreover, use LLMs to filter your data so that it only contains offers inside UE (think of the best way to make the least possible calls!).
2. Build a vectorstore of embeddings and retrieve the best matching job offers (Retrieval Augmented Generation) to the profiles' characteristics (you will find everything you need to know in this short tutorial: https://www.deeplearning.ai/short-courses/langchain-chat-with-your-data)
3. Share the best matching job offers with an LLM and ask them which job offer would be the most appropriate for each profile. The decision must be made based on three factors: the skills required, the role and responsibilities to take on and the benefits the company offers.
4. Justify the answers from the previous step.

You will be scored based on (in order of importance):

1. Your code cleanup, comments, code reproducibility, and using good programming practices.
2. A clear visualization of the whole pipeline.

Please note that finding the best solution will not be the primary scoring criteria. Instead, we are looking for clear and well-documented code, an understanding of the problem, and a reproducible pipeline developed to solve the problem.