Task: Resume Matching with Job Descriptions Using PDF CVs

Objective: Build a PDF extractor to pull relevant details from CVs in PDF format, and match them against the job descriptions from the Hugging Face dataset.

This task outline serves as a general guide, and the tools or models recommended are only suggestions. You are encouraged to leverage any open-source tools, AI models, or language processing techniques you're familiar with or find suitable for the task. We value innovation, creativity, and problem-solving. So, if you believe there's a better approach or a more efficient tool to accomplish this task, please feel free to use it. Document any deviations from the recommendations, and provide a rationale for your choices in the submission.

---

1. PDF Data Extraction

Objective: Extract details from CVs in PDF format.

- Dataset: Kaggle Resume Dataset

Instructions:

Download the Kaggle "resume dataset".
Build a PDF extractor using Python, leveraging libraries such as PyPDF2 or PDFMiner.
Extract the key details:

- Category (Job role)
- Skills
- Education (Degree, Institution)

---

2. Job Description Data Understanding

Objective: Fetch and comprehend job descriptions from the Hugging Face dataset.

- Dataset: Job Descriptions from Hugging Face

Instructions:

Use the Hugging Face datasets library to fetch the job descriptions. For this task, consider extracting 10-15 job descriptions.

---

3. Candidate-Job Matching

Objective: Match extracted CV details against the fetched job descriptions based on skills and education.

Tools Suggested: Use the Transformers library by Hugging Face. BERT or DistilBERT can be a starting point for embedding extraction.

Instructions:

Tokenize and preprocess both the job descriptions and the extracted CV details from the PDFs.

Convert the tokenized text into embeddings using a pretrained model like DistilBERT from Hugging Face.

For each job description, calculate the cosine similarity between its embedding and the embeddings of the CVs.

Rank CVs based on this similarity for each job description.

List the top 5 CVs for each job description based on the highest similarity scores.

Submission:

Submit your code/scripts, including the PDF extractor.

Provide a short report detailing:

- Your approach to the task.
- Challenges faced and solutions.
- Top 5 candidates for each job description based on similarity scores.

Recommendations or insights from the matching process.

Evaluation Criteria:

- Effectiveness of the PDF extraction tool.
- Accuracy of extracted data from the Kaggle dataset.
- Efficient use of pre-trained models for embeddings and similarity.
- Clarity in the approach and documentation.
- Code quality and readability.

Tips for the Intern:

- Start with a small set of PDFs to test your extractor.
- Once the extractor is functioning, expand to the full dataset.
- For cosine similarity calculations, consider using libraries like scikit-learn.
- The approach and problem-solving method are more important than a perfect match. Document any challenges faced.

Once you have completed the task, please upload it to your GitHub and submit the work via the following link-

Deadline: You must submit your work before **19th September 5pm IST** to be considered for the next round.