

Project Description: Resume Matching with Job Descriptions

Executive Summary:

The project aimed to create an automated resume matching system that could efficiently match resumes to job descriptions. By leveraging Natural Language Processing (NLP) techniques and machine learning, this system successfully extracted key information from resumes in PDF format, tokenized and embedded the text, and calculated cosine similarity scores to rank the top matching resumes for each job description.

Description:

This project comprises of 2 parts in the first part we took the single resume in PDF format. In the first step, which is PDF Data Extraction. In this step, we'll extract details from CVs in PDF format using Python and libraries like PyPDF2 or PDFMiner. Here we Extracted Text Cleaning, Extract Key Details and we repeated it for multiple resumes.

Whereas in the second part we took the complete dataset of resumes from kaggle and did the PDF Data Extraction for the major columns.

In the second step we did the Job Description Data Understanding, fetch and comprehend job descriptions from the Hugging Face dataset. We use the load_dataset function from the Hugging Face datasets library to load job descriptions. The Hugging Face library provides access to various datasets, and you can select a suitable one.

Once we have loaded the dataset, we can access the job descriptions. The exact structure of the dataset may vary depending on the dataset we have chose. Typically, we can access job descriptions using the dataset["train"] or dataset["test"] keys, depending on how the dataset is organized.

Now, let's move on to the third step, which is Candidate-Job Matching. In this step, we will match the extracted CV details against the fetched job descriptions based on skills and education using NLP techniques. Here we do the several steps like Tokenization and Preprocessing then we will use a pre-trained model like DistilBERT from Hugging Face to convert the tokenized text into numerical embeddings. After that we calculated the cosine similarity between the embeddings of job descriptions and CVs.

After processing all job descriptions and CVs, we convert the lists of embeddings into matrices using torch.cat.

Finally, we calculate the cosine similarity between all job descriptions and CVs, resulting in the similarity_matrix.

Now in the 4th step, we will analyze the similarity matrix to find the best matches for each job description. This involves sorting the rows (job descriptions) based on the similarity scores for each CV and selecting the top N candidates for each job.

Database used: <https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset/code>