**Natural Language Processing Powered Resume Matching and Candidate Ranking for Recruitment**

Contents

[**1. Project Overview** 1](#_Toc195909257)

[2. System Modules 2](#_Toc195909258)

[2.1. Resume Preprocessing 2](#_Toc195909259)

[2.2. Skill, Education, and Experience Extraction 2](#_Toc195909260)

[2.3. Feature Extraction 2](#_Toc195909261)

[2.4. Resume Classification 2](#_Toc195909262)

[2.5. Semantic Similarity Matching 3](#_Toc195909263)

[3. Streamlit Web Application 3](#_Toc195909264)

[3.1. Features 3](#_Toc195909265)

[3.2. Interface Overview 3](#_Toc195909266)

[4. Deployment & Hosting 3](#_Toc195909267)

[4.1. Local Deployment 3](#_Toc195909268)

[4.2. Cloud Hosting (Optional) 3](#_Toc195909269)

[5. Unit Testing 4](#_Toc195909270)

## **1. Project Overview**

This project focuses on building an intelligent **Natural Language Processing Powered Resume Matching and Candidate Ranking for Recruitment**. It aims to extract structured information from unstructured resumes and match them to job descriptions using classification and semantic similarity methods.

## 2. System Modules

### 2.1. Resume Preprocessing

* **Objective**: Clean and normalize resume text for analysis.
* **Process**:
  + Remove special characters and digits.
  + Normalize white spaces.
  + Convert text to lowercase.

### 2.2. Skill, Education, and Experience Extraction

* **Objective**: Extract relevant entities from resumes.
* **Methods**:
  + **Skills**: Identified using a custom keyword-matching approach with an extensive skill list.
  + **Education**: Extracted by recognizing keywords like degree names (e.g., B.Tech, MBA).
  + **Experience**: Parsed from duration mentions or patterns such as “X years”.

### 2.3. Feature Extraction

* **TF-IDF Vectorization**:
  + Converts resumes and job descriptions into numerical vectors based on word importance.
* **SBERT Embeddings**:
  + Semantic vector representation using Sentence-BERT to capture contextual meaning.

### 2.4. Resume Classification

* **Model Used**: Logistic Regression.
* **Input Features**: TF-IDF vectors / SBERT embeddings.
* **Output**: Predicted job category for each resume or Job Description.

### 2.5. Semantic Similarity Matching

* **Objective**: Rank resumes by relevance to a job description.
* **Method**: Cosine similarity between SBERT embeddings of resumes and the given job description.

## 3. Streamlit Web Application

### 3.1. Features

* Upload and process multiple resumes (PDFs).
* Upload job description in PDF
* Predict job description & Resume category.
* View top matching resumes with Candidate Ranking.

### 3.2. Interface Overview

* **Main Panel**: Upload Job Description (Single) PDF and Resume (Multiple)files
* **Main Panel**: Shows predictions category and top matches with similarity scores.

## 4. Deployment & Hosting

### 4.1. Local Deployment

* Requirements: Python 3.10+, Streamlit, scikit-learn, sentence-transformers, pandas, numpy, PyMuPDF.

**Run using:**  
 streamlit run app.py

* **streamlit run app.py**

### 4.2. Cloud Hosting (Optional)

* Platforms: Streamlit Cloud, AWS, or GCP.
* Needs secure file handling and data privacy considerations.

## 5. Unit Testing

⚡ main ~/Resume-Matching-and-Candidate-Ranking**# python -m unittest tests/test\_preprocessing.py**

[nltk\_data] Downloading package punkt to

[nltk\_data] /teamspace/studios/this\_studio/nltk\_data...

[nltk\_data] Package punkt is already up-to-date!

[nltk\_data] Downloading package stopwords to

[nltk\_data] /teamspace/studios/this\_studio/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

[nltk\_data] Downloading package wordnet to

[nltk\_data] /teamspace/studios/this\_studio/nltk\_data...

[nltk\_data] Package wordnet is already up-to-date!

[nltk\_data] Downloading package punkt\_tab to

[nltk\_data] /teamspace/studios/this\_studio/nltk\_data...

[nltk\_data] Package punkt\_tab is already up-to-date!

----------------------------------------------------------------------

Ran 7 tests in 1.168s

OK

⚡ main ~/Resume-Matching-and-Candidate-Ranking# **python -m unittest tests/test\_extract\_skills\_edu\_exp.py**

----------------------------------------------------------------------

Ran 2 tests in 0.005s

OK

⚡ main ~/Resume-Matching-and-Candidate-Ranking# **python -m unittest tests/test\_feature\_extraction.py**

Batches: 100%|██████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 81.73it/s]

Batches: 0it [00:00, ?it/s]

Batches: 100%|██████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 70.02it/s]

Batches: 100%|██████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 78.02it/s]

Batches: 100%|██████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 82.45it/s]

----------------------------------------------------------------------

Ran 10 tests in 2.660s

OK

⚡ main ~/Resume-Matching-and-Candidate-Ranking# **python -m unittest tests/test\_classification.py**

----------------------------------------------------------------------

Ran 2 tests in 0.801s

OK

⚡ main ~/Resume-Matching-and-Candidate-Ranking# **python -m unittest tests/test\_match\_score.py**

Batches: 100%|██████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 64.85it/s]

Batches: 100%|██████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 42.69it/s]

----------------------------------------------------------------------

Ran 2 tests in 1.045s

OK