Resume Analyzer with Job Matching

CECS 218 Final Project Report

Team Members: Nasser Al-qarni & Albara Al-saab

Date: 2025/05/07

Resume Analyzer with Job Matching

Tools and Technologies Used:

- **Python 3.11**
- **Regular Expressions**
- **PyPDF2 and python-docx for reading resume files**
- **scikit-learn for text vectorization and cosine similarity**
- **matplotlib for visualization**
- **Exception Handling and Modular OOP Design**

Resume Analyzer with Job Matching

System Modules

Resume_reader:

Handles reading and parsing resumes from both .pdf and .docx formats.

- _extract_text_pdf / _extract_text_docx: Load and clean file text.
- extract_email: Uses regex to find a valid email address.
- extract_name: Assumes the first non-empty line is the candidate's name.
- extract_skills: Checks for known skill keywords in resume text.

job_matcher:

Matches the candidate's skills to predefined job profiles:

- Job roles include Data Scientist, Web Developer, Android Developer, and DevOps Engineer.
- Uses CountVectorizer and cosine_similarity to find the closest job match.
- Returns the best match and similarity score.

Visualization

Generates a horizontal bar chart showing similarity scores for all job roles to help users interpret matching strength visually.

Resume Analyzer with Job Matching

Code Architecture:

sample_cvs/

└── resume1.pdf ... to resume10.docx

main.py ← Program entry point

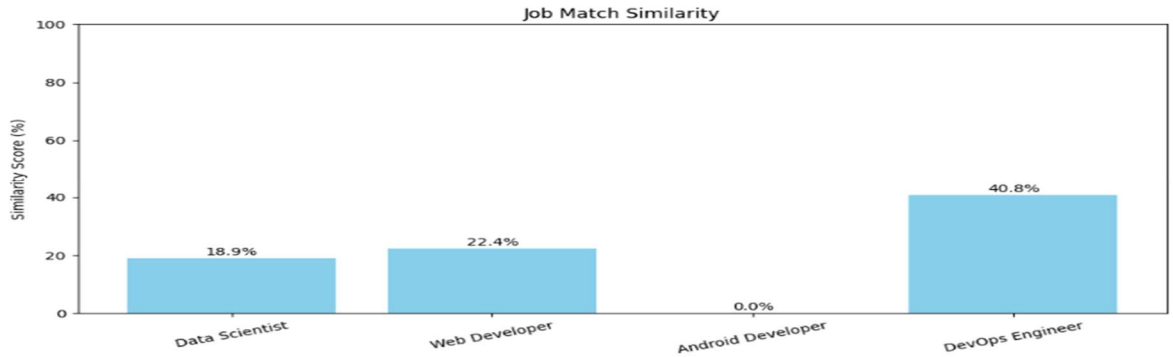
resume_reader.py ← Resume parsing logic

job_matcher.py ← Job matching + visualization

Example output:

```
Name: Tom Brooks
Email: tom.brooks@example.com
Extracted Skills: ['python', 'react', 'docker', 'aws']
Best Job Match: DevOps Engineer (Similarity Score: 0.41)
```

Bar Chart:



Resume Analyzer with Job Matching

Challenges Faced:

- **Extracting consistent data from resumes with non-standard formatting.**
- **Ensuring skill detection was accurate but not overly rigid.**
- **Gracefully handling missing or unsupported file formats.**
- **Visualizing and debugging similarity scores during early testing.**

Conclusion:

This project demonstrated how Python can be used to automate resume parsing and job matching through modular design and machine learning-based similarity. It successfully implements OOP principles, file handling, and visualization.