

# **SmartResumeAI**

## **Intelligent Resume Reviewer using LLMs and Azure**

### **Overview**

In the competitive job market, understanding how well a candidate's resume aligns with a job description is crucial not just for employers but also for job seekers. Traditional resume screening methods, typically employed by companies, rely heavily on keyword-based filtering systems. While useful, these systems often miss the subtleties in resumes and job descriptions, potentially leading to overlooked opportunities and a lack of constructive feedback for candidates.

This project addresses the needs of job seekers who want to proactively assess their resumes against job descriptions, identify gaps, and receive personalized insights on how to improve. By leveraging Large Language Models (LLMs), our AI-powered resume reviewer goes beyond simple keyword matching to understand the context and content of both resumes and job descriptions. This enables the system to provide more accurate matching and detailed suggestions for enhancement.

The project will allow users to upload job descriptions and resumes, generate embeddings for both, and calculate similarity scores using cosine similarity. Beyond matching, the system will analyze the resume against the job description to highlight areas for improvement, offering suggestions such as relevant skills, action words, and formatting tips. Finally, the application will be deployed on Azure, ensuring it is scalable and accessible to users seeking to refine their resumes and improve their job prospects.

### **Aim**

The aim of this project is to develop and deploy an AI-powered system that enables users to assess how well their resumes match job descriptions, identify gaps, and receive personalized suggestions for improvement to enhance their job prospects.

### **Data Description**

The dataset for this project consists of unstructured text data, including job descriptions and resumes in PDF or image format. The resumes and job descriptions are parsed to extract relevant textual information for analysis.

### **Tech Stack**

- Language: Python 3.10
- Libraries: Langchain, Langchain-openai, streamlit, tesseract
- Model: Open AI Embeddings, GPT-4
- Cloud Platform: Microsoft Azure

## Approach

- **Data Parsing and Preparation**

- Use PDF/Text parsers to extract text from resumes and job descriptions.
- Preprocess the text data to remove any irrelevant information and

- standardize the format. ● **Embedding Generation**

- Generate embeddings for both resumes and job descriptions using pre-trained models like OpenAI embeddings.
- Ensure embeddings capture the semantic meaning of the text for accurate

- similarity matching. ● **Similarity Calculation**

- Use cosine similarity to calculate the similarity scores between resumes and job descriptions.
- Rank resumes based on their similarity scores to identify the best matches.

- **Gap Analysis**

- Analyze the differences between resume content and job requirements.
- Identify key areas where the resume does not align with the job description, such as missing skills or irrelevant experience.

- **Suggestions Generation**

- Generate personalized suggestions for resume improvement using Large Language Models (LLMs).
- Provide actionable recommendations like relevant skills, action words, and formatting tips.

- **User Interface Development**

- Build a frontend UI using Streamlit to allow users to upload resumes and job descriptions.
- Display similarity scores, gap analysis, and improvement suggestions in an easy-to-understand format.

- **Feedback Loop and Continuous Improvement**

- Implement a feedback mechanism where users can rate the suggestions provided.
- Use this feedback to refine the prompts and improve the accuracy and relevance of suggestions.

- **Deployment**

- Deploy the application on Azure using Azure App Service.

- Ensure scalability and accessibility for users, allowing them to assess and improve their resumes efficiently.

### Modular code overview:

Once you unzip the modular\_code.zip file, you can find the following:

```
├─ jd_data/
├─ output
│   └─ jd_embeddings_large.pkl
│       └─ resume_embeddings_large.pkl
├─ readme.md
├─ requirements.txt
├─ resume_data/
├─ Resume_Scorer.ipynb
├─ Resume_Suggestions.ipynb
└─ src
    ├── constants.py
    ├── directory_reader.py
    ├── embedding_model.py
    ├── resume_scorer.py
    └─ resume_suggestions.py
```

### Project Takeaways

1. Understand the challenges of traditional resume screening and how AI can enhance the process
2. Learn the importance of semantic understanding in matching resumes with job descriptions
3. Gain hands-on experience with text parsing and data preprocessing for unstructured data like resumes and job descriptions
4. Explore the concept of embeddings and how they represent the meaning of text data
5. Learn to generate and use embeddings for similarity matching between different textual inputs
6. Understand cosine similarity and how it's used to measure the similarity between resumes and job descriptions

7. Learn to identify gaps in resumes by comparing them to job descriptions and analyzing key areas for improvement
8. Discover how to use Large Language Models (LLMs) to generate personalized suggestions for resume enhancement
9. Get familiar with prompt engineering and how to design effective prompts for various tasks
10. Understand the fundamentals of Langchain, including prompt chaining, memory, and conversation handling
11. Develop skills in building a user-friendly web application using Streamlit for resume analysis
12. Learn how to implement a feedback loop to continuously improve the system based on user input
13. Gain insights into deploying AI applications on Azure using Azure App Service, including setup and configuration