

AI Recruitment Tool

1. Introduction

Objective: The design and implementation of a recruitment tool that leverages LLMs to enhance the hiring process through job matching and interview analysis.

Goal: Develop a job matching system that uses AI to analyze candidate profiles and job descriptions. The system will automatically match candidates to job openings based on key data points like skills, work experience, and qualifications. It will provide suitable job recommendation for candidates, utilizing LLM, vector databases and Retrieval-Augmented Generation (RAG) for efficient and accurate matching. Analyze video interviews for accuracy and communication skills, providing detailed feedback.

2. Project Overview

The recruitment tool consists of two primary modules:

- **Job Matching System:** Automates the alignment of resumes with job descriptions using AI and vector databases.
- **Interview Analysis Module:** Analyzes video interviews for communication skills and provides feedback.

3. Design Choices

3.1 Architecture

- The system is designed using a modular architecture to facilitate maintainability and scalability.
- A microservices approach allows for separation of concerns between the job matching and interview analysis components.

3.2 Technologies Used

- **Frontend and Backend:** Streamlit for creating an interactive web application.
- **AI/ML:** Hugging Face's API inference to make calls to LLMs.
- **Vector Database:** FAISS for efficient similarity search on embeddings.
- **Video Transcribe:** moviepy and speech_recognition by Google to transcribe interview recordings.

3.3 Data Flow

1. **Job Description and Resume Upload:** Users upload job descriptions and resumes, which are parsed to extract key features.
2. **Embedding Generation:** The uploaded content is converted into embeddings using the selected embedding model.
3. **Document Storage:** Embeddings are stored in a vector database for efficient retrieval.
4. **Querying:** When a candidate's resume is uploaded, it is matched against stored job descriptions to identify suitable roles.
5. **Interview Analysis:** Video interviews are processed into text; it is then summarized using a text summary model to access candidate's communication skills.

4. Implementation Choices

4.1 Job Matching System

- **Parsing:** Utilized NER techniques to extract relevant information from job descriptions and resumes.

NER Model – bert-large-cased-finetuned-conll03-english

- **Vector Storage:** Implemented a custom VectorStore class to manage embedding storage and retrieval.

Vectore DB – Faiss index

- **Create Embedding:** utilized embedding LLM to convert the parsed resume and job descriptions into vectors for storage and match scoring.

Embedding Model - sentence-transformers/all-MiniLM-L6-v2

- **Retrieval-Augmented Generation (RAG):** Employed to enhance the response generation based on retrieved documents.

4.2 Interview Analysis Module

- **Video Processing:** Designed a module to handle video uploads and convert them into text.

Video to audio - moviepy.editor

Audio to text - speech_recognition

- **Feedback Mechanism:** Developed a scoring system to evaluate and provide feedback on candidates' performance by summarizing the text.

LLM - Mistral-7B-Instruct-v0.3

5. Assumptions

The quality of the uploaded video interviews is sufficient for analysis, the implementation is not very fine tuned and hence will summarize any given video. Because direct LLM call is made.

Job descriptions and resumes are formatted consistently and have all the key sections named like Skills, years of exp, education, certifications, etc. Because NER model is used.

6. Prompts

Pre built entities for NER model –

```
extracted_info = {  
    "skills": [],  
    "years_of_experience": [],  
    "education": [],  
    "certifications": [],  
    "technologies": []  
}
```

Prompt to summarize the interview video –

Please summarize the following interview transcript in the categories provided below. Do not include any part of the prompt in your response.\n\n"

```
f"***Transcript:**\n{transcript}\n\n"
```

```
f"***Please provide your analysis as follows:**\n"
```

```
f"- **Communication Style**:\n"
```

```
f"- **Active Listening**:\n"
```

```
f"- **Engagement with the Interviewer**:\n"
```

```
f"***End of instructions.**"
```