

AI Interview Guru

In HR business units, a common challenge lies in streamlining the hiring process. On average, thousands of resumes flood in for a single position, requiring team members to carefully scan through each one to find the best fit for the job. This task is not only time-consuming but also leads to prolonged vacancies and increases the risk of unconscious biases.

To address these issues, a job match tool can streamline this process. By automatically matching candidates' skills and ranking them based on their alignment with job requirements in a matter of seconds, this tool eliminates the need for extensive manual effort from the HR department. Consequently, HR professionals can redirect their focus to more strategic aspects of the hiring process, ensuring they can identify and hire the most suitable candidate efficiently.

Welcome to AI Interview Guru, your ultimate tool for streamlining the job application process and enhancing interview preparation using cutting-edge artificial intelligence technologies.

Solution

The AI Interview Guru is a user-friendly web app created to help analyze candidates' resumes compared to job descriptions. This idea of matching resumes to jobs went through three stages of development using modeling techniques. This ensured we selected the best solution with the highest accuracy for its intended purpose.

1. **FIRST STEP: CLASSICAL NLP TECHNIQUES - PARSING MODEL:** This step involved leveraging classical NLP techniques like text tokenization, stop word removal, Named Entity Recognition, Regular Expressions, and skill extraction. We utilized these methods to build a model capable of matching candidate skills with job requirements. (For more details, refer to further documentation.)
2. **SECOND STEP: BEST MODEL EVALUATION – LLM COMPARISON:** In this phase, we evaluated five models using LLM and pre-processing techniques. Each model underwent rigorous testing with the same dataset to determine the most suitable one for production. (For more insights, see further documentation.)
3. **THIRD STEP: PRODUCTION MODEL - AI INTERVIEW GURU DEPLOYMENT:** After thorough experimentation to identify the optimal model, we deployed the chosen one using Streamlit. This deployment allowed us to conduct real-world tests with candidates, refining the performance of the AI Interview Guru. (For additional information, consult further documentation.)

Conclusions

As a consulting group, our assessment indicates that implementing the AI Interview Guru has the potential to enhance hiring quality, reduce cost per hire, expedite time-to-hire, and mitigate bias in decision-making. By utilizing a combination of classic and advanced techniques during research, we arrived at a more precise decision for the model deployed in production. Moving forward, it is recommended to monitor and supervise the tool's performance to gather results that can be leveraged for training a more robust platform in the future.

FIRST MODEL: CLASSICAL NLP TECHNIQUES - PARSING MODEL IN STREAMLIT

Overview

This first version was a Streamlit job matching system, a user-friendly web application designed to assist in analyzing candidates' CVs (Curriculum Vitae) against job descriptions to determine the level of skill match.

Pros and Cons

- Pros: this version first version was able to extract the skills from CVs and Job Postings and do the crossing given a corpus with listed skills. With this model we ensured a literal accuracy avoiding hallucination and randomly assigned scores, but a probabilistic data-driven model.
- Cons: limited corpus with listed skills vs GPT models that provide further capabilities.

Functionality: candidate's info and skills extraction, job posting required skills extraction and comparison and scoring skills matching between candidates and job postings.

Techniques: text tokenization, stop word removal, Name Entity Recognition, Regular Expressions and Extract skills.

- Text Tokenization: Breaks down CV text into individual words/phrases for easier processing.
- Stop Word Removal: Removes common words (e.g., "the", "a") to focus on relevant keywords and skills.
- Named Entity Recognition (NER): Identifies and classifies named entities like names and contact information in the CV.
- Regular Expressions: Uses patterns to specifically extract email addresses and phone numbers from the text.
- Extract_skills: extracts skills from a CV and job posting by matching individual words (one-grams) and noun phrases (bigrams) against a predefined list, returning a unique and capitalized list.

Supported formats: pdf and txt.

Libraries used:

- streamlit: Creates a web app interface.
- re: Regular expressions for text processing.
- spacy: Natural Language Processing library for named entity recognition and skill extraction.
- fitz: Processes PDF documents for text extraction.
- pandas: Data manipulation and table creation.

SECOND STEP: BEST MODEL EVALUATION – LLM COMPARISON:

In this phase, we evaluated five models using LLM and pre-processing techniques. Each model underwent rigorous testing with the same dataset to determine the most suitable one for production.

Main Models

The following models were used to preprocess text

1. **ChatCompletion Model:** The ChatCompletion model, powered by OpenAI's GPT-3.5, is used for **Preprocessing job descriptions and resumes, parsing resumes to extract relevant insights**

The following models were used to assess the best model for production

Sentence Transformer Models

1. **Hugging Face all-mpnet-base-v2:** This is a sentence-transformers model. It maps sentences & paragraphs to a 768-dimensional dense vector space and can be used for tasks like clustering or semantic search.
2. **Hugging Face paraphrase-MiniLM-L6-v2:** This is a sentence-transformers model. It maps sentences & paragraphs to a 384-dimensional dense vector space and can be used for tasks like clustering or semantic search.
3. **Hugging Face all-MiniLM-L12-v1:** This is a sentence-transformers model: It maps sentences & paragraphs to a 384-dimensional dense vector space and can be used for tasks like clustering or semantic search.
4. **Hugging Face multi-qa-mpnet-base-dot-v1 :** This is a sentence-transformers model: It maps sentences & paragraphs to a 768 dimensional dense vector space and was designed for semantic search.

Embedding Vector

5. **Open AI Embedding Vector Model:** This model, **text-embedding-ada-002, text-embedding-3-small, text-embedding-3-large**, is tailored for **generating embeddings**, which are **numerical representations of textual data**. Embeddings capture the semantic meaning of words, phrases, or entire documents in a high-dimensional vector space.

Scoring Explanation

We utilize cosine similarity to assess the similarity between resumes and job descriptions. Cosine similarity calculates the angle between these vectors, indicating how closely they align.

$$\text{score} = \sum_{i=0}^N \max(\text{cosine_similarity}(\mathbf{A}_i, \mathbf{B}))$$

Results

Based on our comprehensive analysis, we found that all models exhibit a similar level of semantic understanding. However, the size of the models emerged as a crucial factor in selection due to the higher computational resources required for inference compared to smaller counterparts. Furthermore, an additional experiment involving semantic similarity combined with our previous parsing techniques revealed that "text-embedding-3-small" performed comparably to larger models while consuming fewer resources. This finding underscores the efficiency and effectiveness of smaller models in achieving comparable performance with reduced resource consumption.

THIRD STEP: PRODUCTION MODEL - AI INTERVIEW GURU DEPLOYMENT:

After thorough experimentation to identify the optimal model, we deployed the chosen one using Streamlit. This deployment allowed us to conduct real-world tests with candidates, refining the performance of the AI Interview Guru.

Main Models

2. ChatCompletion Model

The ChatCompletion model, powered by OpenAI's GPT-3.5, is used throughout AI Interview Guru to assist HR recruiters and job seekers in **Preprocessing job descriptions and resumes, parsing resumes to extract relevant insights**

6. Embedding Vector Model

This model, text-embedding-3-small, is tailored for **generating embeddings**, which are **numerical representations of textual data**. Embeddings capture the semantic meaning of words, phrases, or entire documents in a high-dimensional vector space.

Scoring Explanation

Cosine Similarity Score

The cosine similarity score measures the cosine of the angle between two vectors in a multidimensional space. In the context of AI Interview Guru, it quantifies the similarity between the embeddings of a resume and a job description. The formula for cosine similarity is as follows:

Where:

- A and B are the embedding vectors of the resume and job description, respectively.
- $|A|$ and $|B|$ represent the magnitudes (or norms) of the vectors.

Usage

JobMatchRecruiter: Upload resumes and job descriptions to match candidates with suitable positions. Receive detailed scoring and insights to make informed hiring decisions.

The app is ready to use in the following [Link](#)

Installation

To run AI Interview Guru locally, follow these steps:

1. Clone the repository to your local machine.
2. Install the required dependencies using `pip install -r requirements.txt`.
3. Set up your OpenAI API key as a secret variable in Streamlit.

4. Run the application using streamlit run AI_interview_checker.py.
5. Access the application through the provided URL.

Contributing

This project was part of NLP course in ie University, Master of Business Analytics and Big Data program. This project was developed by @ Contributions to AI Interview Guru are welcome! If you have suggestions for new features, improvements, or bug fixes, please submit a pull request or open an issue on GitHub.