Job Skill Analyses using Generative AI models and NER(NLP)

# Introduction

In the rapidly evolving job market, staying updated with the latest technical skills is crucial for both job seekers. Analyzing job descriptions can provide valuable insights into the latest trends and skills required across various industries. This project aims to utilize LangChain, OpenAI's API, natural language processing (NLP), specifically Named Entity Recognition (NER), to identify the technical skills from job descriptions and get the overview of the current market

# Problem Statement

The primary challenge addressed the dynamic nature of job market requirements. Traditional methods of skill identification often lag behind, failing to capture the most recent trends and demands. Job descriptions, which are abundant on the web, are a valuable source of information but are often unstructured and voluminous, making manual analysis impractical.

The goal of this project is to develop an automated system that can efficiently scrape job descriptions from the web using BeautifulSoup, and then apply LangChain, OpenAI's API and NLP techniques to extract relevant technical skills. The goal of the project is to

1. **Web Scraping:** Use BeautifulSoup to gather job descriptions from online job portals.
2. OpenAI's API**:** Use OpenAI's API to identify and extract technical skills from the scraped job descriptions.
3. **NLP Techniques:** Use NER to identify and extract technical skills from the scraped job descriptions.
4. **Analyze Trends:** Provide insights into the latest technical skills in demand, helping job seekers to stay competitive and employers to align their hiring strategies with current market needs.

# Methodology

To achieve the goal of identifying the latest technical skills from job descriptions, the methodology involves several key steps, including web scraping, data cleaning, and the application of advanced NLP techniques using LangChain and OpenAI's API.

### 1. Web Scraping

Using BeautifulSoup, job IDs and corresponding job descriptions are scraped from linkedin. This involves extracting relevant data, and storing it in a structured format for further processing.

### 2. Data Cleaning

The scraped job descriptions are loaded into a data processing framework such as pandas. The data is then cleaned by removing null or incomplete values.

### 3. (First Approach) Skill Extraction Using LangChain and OpenAI API

#### Chunking Data:

Divide the cleaned job descriptions into smaller chunks to reduce token size and facilitate processing.

##### Vector Embeddings:

Convert the text chunks into vector embeddings using OpenAIEmbeddings. This process transforms textual data into numerical representations that capture semantic meaning.

##### Vector Storage:

Store the vector embeddings in a vector database such as Pinecone. This allows for efficient retrieval and similarity search.

#### Prompt Similarity search**:**

Perform a cosine similarity search in the vector database to retrieve the most relevant job description chunks.

#### Skill Extraction:

Use the OpenAI API to analyze the retrieved job description chunks and extract technical skills. This involves querying the model to identify skills based on the content of the job descriptions.

### 4. Second Approach: Skill Extraction Using NER, NLP, and OpenAI API

#### **Create Custom SpaCy Model:**

A blank SpaCy model called "nlp" is created, and a custom entity ruler with the label SKILL as the entity is added. The list of skills from "skills.csv" is loaded into this model to facilitate initial skill identification.

#### **Create Training Data and Train a New SpaCy Model:**

To train a model capable of identifying new skills, the existing model is used to identify skills from the job descriptions, which are then formatted according to SpaCy's requirements to serve as training data. A new blank SpaCy model called "nlp2" is trained using this data. The pipelines from "nlp" and "nlp2" are combined to create the NER\_Skill\_Final model. This final model is used to extract a list of skills from job descriptions, and a dictionary is created to store the skills along with their respective counts.

#### **OpenAI API to Refine the Results:**

The dictionary of skills is refined by using the ChatOpenAI model (model="gpt-4") to extract relevant technical skills from the already concise data. This step ensures that the extracted skills are accurate and relevant to the job description.

# **Result of the First Approach:**

The OpenAI API model "gpt-3.5-turbo-instruct" exhibited hallucination behavior, generating skills not present in the given job descriptions. This issue persisted even with the temperature parameter set to 0, indicating a challenge in accurately extracting relevant skills without introducing erroneous data.

# **Result of the Second Approach:**

The combination of NER and NLP allows for the identification of existing skills from job descriptions. Since the data is already concise, the OpenAI API can extract relevant information without hallucinating. The NER\_Skill\_Final SpaCy model is also capable of identifying new skills, leading to a more comprehensive extraction of technical skills from job descriptions.

# Future Scope

To enhance the accuracy and robustness of the current methodology for identifying technical skills from job descriptions, the model should be trained with more extensive and relevant datasets.

Furthermore, refining the criteria for what constitutes a "skill" will be crucial. As there are no hard and fast rules governing skill identification, the model sometimes includes irrelevant skills and keywords in the output list. Developing a more nuanced understanding and definition of skills within various contexts can help in creating a more precise model.