Project Title :

Job Recommender

Technologies: NLP, Deep Learning

Group 15 : Team members

Bhavesh Kumar Chigullapalli

Bhargav Sai Pendyala

Oruganti Chetan Reddy

Khadyothan Choudari Dasari

Kanala Sai Bhargava Reddy

Problem Statement and Motivation

- Traditionally, job suggestion has been thought of as a filter-based match or a recommendation based on the characteristics of positions and candidates as independent entities.
- We can apply this approach that uses machine learning to utilise the advancement of applicant job selection.
- Additionally, we can propose some suggestions which are made up of multiple recommendations that are beneficial for the end user.
- Our Goal here is to create a machine learning model using NLP techniques to find the best job that is suited for the employee based on his skills/profile.

Supporting articles/papers/open source project for the job recommender project using NLP and Deep Learning.

- Job Recommender systems using NLP
- Training deep Neural Networks
- Neural Architectures for Named Entity Recognition
- Teaching machines to read
- Natural Language Understanding
- Natural Language Processing (almost) from Scratch
- TensorFlow API Documentation
- Keras API reference

Project Modules

- **Web Scraping:**

We will be creating the dataset by scrapping from different platforms like Glass Door, LinkedIn, Naukri etc.

- **Data Cleaning/Transformation:**

We will scrape the raw data using Webdriver and later transform this data into useful format by using preprocessing techniques and NLP techniques.

- **Model Development/Evaluation using Cosine Similarity:**

In this module, we use tf-idf vectorization to vectorize the text in the dataset, and then we apply cosine similarity to find similarity between two texts and forecast the job with highest score.

- **Testing and Deployment:**

In this module we test our model's results and try to improvise the model using parameter tuning.

WORK DISTRIBUTION AND CONTRIBUTION OF TEAM MEMBERS

- ➔ **Setting up project environment and Installing required libraries (Khadyothan and Chetan)**
- ➔ **Finding, Downloading and Modifying Dataset to our needs(Bhargava.K, Bhavesh, Bharghav Sai.P)**
- ➔ **Preprocessing and NLP :**
  - **IT dataset (Bhavesh , Bhargav.K, Khadyothan)**
  - **nonIT dataset (Chetan and Bharghav Sai.P)**
  - **Cosine Similarity (Bhavesh , Bhargava.K, Khadyothan)**
  - **Porter stemmer (Chetan and Bhargav Sai.P)**
- ➔ **Job-recommender model - (Bhavesh , Bharghav Sai.P, Chetan, khadyothan, Bhargava.K)**
- ➔ **Documentation - (Bhavesh)**

Tools and Libraries Used :

Data collection :

- Beautiful Soup(Raw data scraped from indeed.com)

Data cleaning and processing using NLP :

- Natural Language Toolkit (NLTK) and PorterStemmer for Stemming
- Numpy, Pandas, Re.

Model Development:

- Tensorflow
- Keras
- Scikit-learn

Porter Stemmer :

What is Stemming :

- ➔ Stemming is the process of reducing a word to its roots, also known as a lemma. words such as “Likes”, ”liked”, ”likely” and ”liking” will be reduced to “like” after stemming.

Porter Stemmer :

- ➔ It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes.
- ➔ Example :: caresses - caress | relational - relate | ponies - poni .

Why Porter Stemmer :

- ➔ Porter's stemmer advantage is its simplicity and speed. Porter algorithm was made in the assumption that we don't have a stem dictionary and helps improve the performance of our model, significantly at larger scales.
- ➔ Compared to other stemmers porter stemmer has a relatively less error rate.

Cosine Similarity:

Cosine Similarity :

- Cosine similarity measures the similarity between two vectors of an inner product space. It is uses cosine rule. It is often used to measure document similarity in text analysis.

Why Cosine Similarity:

- While comparing two documents, Euclidean distance is misleading where document size increases.

Implementing Cosine Similarity:

- We use tf-idf vectorization to convert the text into vectors.
- Then compare the two vectors using cosine rule
  - Cos (a , b) = (a . b) / (|a| * |b|)
  - Where a.b is the dot product of vectors a and b.

Github link for the project : https://github.com/bhavesh-20/Job-Recommender

Work Plan :