Week 10

Name: Sri Harsha, Nimmagadda
Email: harshachaitanya27@gmail.com
College: Oregon State University
Specialisation: NLP

Problem Description

As a recruiter, we often face the challenge of reviewing thousands, sometimes millions, of resumes for a single position. This overwhelming volume makes it incredibly difficult and time-consuming to thoroughly evaluate each candidate. Many companies have addressed this issue by implementing systems that require applicants to manually re-enter their information from their CVs into designated fields. While this benefits employers, it's frustrating for candidates who spend hours crafting their resumes and cover letters, only to then spend extra time re-entering the same information.

Business Understanding

I've observed that the manual evaluation of resumes, often filled with excess and inconsistent information, can be tedious and inefficient. To streamline this process, I explored automating resume analysis by reading various file formats (CVs) and leveraging Natural Language Processing (NLP) techniques like word parsing, chunking, regex parsing, and Named Entity Recognition (NER). This allowed me to quickly extract key information like name, email, address, education, and experience from a large number of documents.

Project Life Cycle

1. Data Understanding and Exploration: I began by familiarizing myself with the data, which was unstructured text in JSON format. Importing it as a Pandas DataFrame revealed two main columns: "content" (the resume text) and "annotation" (labeling the extracted information). These represented resumes of 200 individuals,categorized into various fields like name, location, contact information, university/college, degree, graduation year, experience, company names, designation, and skills.
2. Data Cleaning and Transformation: I addressed data issues like inconsistent capitalization, contractions,special characters, stop words, and tokenization using standard NLP techniques. For feature engineering, I employed TF-IDF

(Term Frequency-Inverse Document Frequency) with n-grams ranging from 1 to 3.

3. Exploratory Data Analysis (EDA): Initially, I explored various ways to analyze the text data. I checked for duplicate resumes and dropped one. I also created a dictionary to map common contractions to their expanded forms. After removing stop words, I analyzed average word length, word count distribution, and the frequency of unigrams, bigrams, and trigrams. This revealed insights like the most mentioned companies, keywords emphasized by applicants, and potential job role based on these trends. I attempted K-means clustering but found it ineffective due to the high dimensionality of the TF-IDF vectors. Hierarchical clustering, however,successfully grouped the resumes into 8 distinct clusters, showcasing the potential of organizing unstructured text data for recruitment using appropriate feature engineering.
4. Model Building: I trained a NER model using spaCy. The model learned by example, predicting labels for unlabeled text and receiving feedback based on my corrections. This iterative process improved its accuracy.
5. Model Deployment: As requested, I created functions to accept various file types (.pdf, .docx) beyond .txt.Although functional, I wasn't able to fully deploy the application due to technical challenges and time constraints. The model, however, performs well on the notebook and can be tested with sample resumes provided in the repository.

Limitations

Despite successfully creating the model, limitations arose. Firstly, the deployment faced technical hurdles preventing full application development. Secondly, the model exhibited some inaccuracies in information extraction, likely due to the inherent variability in resume formats and the limited dataset used for training. Generating my own labeled data was impractical due to the time and effort required, and publicly available datasets were scarce.