Resume Categorization and Parsing: A Machine Learning Approach

1. Introduction

1.1 Project Overview

The modern recruitment process often involves sifting through large volumes of resumes, a task that is both time-consuming and prone to human error. To address this issue, this project proposes an automated system that leverages machine learning techniques to categorize resumes into predefined job categories and extract key details such as name, contact information, and skills. This system aims to streamline the resume screening process, ensuring more consistent, accurate, and efficient results.

1.2 Motivation

The motivation for this project stems from the need to:

- **Improve Efficiency:** Automation speeds up the resume screening process, allowing HR teams to handle larger volumes of applications.
- **Ensure Consistency:** A machine learning model can standardize the evaluation process, reducing biases and ensuring that all resumes are assessed based on relevant skills and experiences.
- **Enhance Candidate Experience:** Faster processing of resumes results in quicker responses to candidates, improving their overall experience.
- **Provide Scalability:** The system is scalable and can adapt to varying levels of recruitment demand.

1.3 Objectives

The primary objectives of this project are:

- To categorize resumes into predefined job categories using machine learning.
- To extract key details such as the candidate's name, email, phone number, and skills.
- To evaluate resumes based on Applicant Tracking System (ATS) criteria and generate an ATS score.

2. Methodology

2.1 Data Collection

The dataset used in this project comprises resumes in text and PDF formats, categorized into various job roles such as Data Scientist, Java Developer, and HR. The resumes were collected and labelled based on the job roles they pertain to.

2.2 Text Extraction

Text extraction is a critical first step in processing resumes. For PDF resumes, the PyPDF library was utilized to extract the text content. The text was then cleaned and pre-processed to remove unnecessary characters, links, and formatting issues. This cleaned text formed the basis for feature extraction and subsequent classification.

2.3 Feature Engineering

To transform the textual data into a format suitable for machine learning models, the Term Frequency-Inverse Document Frequency (TF-IDF) method was employed. TF-IDF captures the importance of words within a document relative to their occurrence in the entire dataset. This method reduces the influence of common terms and highlights terms that are more relevant to the classification task.

2.4 Model Selection and Training

The project employs the K-Nearest Neighbours (KNN) algorithm for classification, wrapped within a One-vs-Rest (OvR) strategy. The OvR strategy decomposes the multi-class classification problem into multiple binary classification tasks, allowing the KNN model to predict the most suitable job category for each resume. The model was trained on a labelled dataset, ensuring that it could accurately classify resumes into predefined categories.

2.5 Category Prediction

Once trained, the model is used to predict the job category of new resumes based on the text features extracted. The model outputs the most likely job category, allowing for efficient categorization of resumes.

2.6 ATS Score Calculation

The system also evaluates each resume's alignment with ATS criteria, generating a score based on the presence of relevant keywords, skills, and information typically prioritized by ATS software. This score helps determine how likely a resume is to pass through an ATS and reach a human recruiter.

2.7 Key Details Extraction

In addition to categorization, the system extracts key details from each resume, including the candidate’s name, email address, phone number, location, college name, and skills. This information is crucial for further analysis and decision-making during the recruitment process.

3. Implementation

3.1 Technology Stack

The project is implemented using Python, chosen for its rich ecosystem of libraries and tools for data processing and machine learning.

Libraries Used:

- **NLTK:** For text processing and natural language understanding.
- **Scikit-learn:** For implementing the machine learning model, including the KNN classifier and the OvR strategy.
- **PyPDF:** For extracting text from PDF resumes.

3.2 Code Structure

The project code is divided into several modules, each handling a specific task in the resume processing pipeline:

1. **Text Extraction Module:** Extracts and cleans text from resumes.
2. **Feature Engineering Module:** Transforms text into numerical features using TF-IDF.
3. **Model Training Module:** Trains the KNN model using the labeled dataset.
4. **Prediction Module:** Predicts the job category of new resumes.
5. **ATS Scoring Module:** Calculates the ATS score based on relevant criteria.
6. **Details Extraction Module:** Extracts key information such as name, contact details, and skills from the resumes.

3.3 Example Workflow

1. **Input:** A resume in PDF format is uploaded.
2. **Processing:** The text is extracted, cleaned, and transformed into TF-IDF features.
3. **Prediction:** The trained KNN model predicts the job category.
4. **Scoring:** The ATS score is calculated.
5. **Output:** The system outputs the predicted category, ATS score, and extracted details.

4. Results

4.1 Model Performance

The KNN classifier, when trained on the labeled dataset, showed satisfactory performance in predicting the job categories of resumes. The accuracy of the model varies depending on the complexity and variety of the resumes within each category.

4.2 Categorization Results

The system was able to accurately categorize resumes into several predefined job roles. The most frequently predicted categories included:

- Java Developer: 84 resumes
- Testing: 70 resumes
- DevOps Engineer: 55 resumes
- Python Developer: 48 resumes

4.3 ATS Scoring

The ATS scoring mechanism was successfully implemented, providing a score that reflects the resume's alignment with typical ATS criteria. This score helps recruiters quickly assess the relevance of each resume.

4.4 Details Extraction

The system accurately extracted key details from the resumes, providing structured information that is useful for further analysis and decision-making.

5. Conclusion

5.1 Summary

The Resume Categorization and Parsing project demonstrates the effectiveness of machine learning in automating the resume screening process. By categorizing resumes into predefined job roles, extracting key details, and calculating ATS scores, the system significantly improves the efficiency and accuracy of the recruitment process.

5.2 Future Work

Future enhancements to the system could include:

- **Improved Model Accuracy:** Further refinement of the model, possibly by incorporating more advanced techniques such as deep learning, could improve classification accuracy.
- **Category Expansion:** The system could be expanded to include a wider range of job categories.
- **Multilingual Support:** Adding support for resumes in multiple languages would broaden the system's applicability.
- **HR System Integration:** Integrating the system with existing HR software would provide a seamless experience for recruitment teams.