RESUME CATEGORIZATION APPLICATION

Submitted for

Statistical Machine Learning CSET211

Submitted by:

(E23CSEU1150) V S SANTHOSH

Submitted to

AMIT SONI

July-Dec 2024

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INDEX

Abstract

The **Resume Categorization Application** is designed to streamline recruitment processes by automating the categorization of resumes into predefined job roles or skill sets. Leveraging **Natural Language Processing (NLP)** techniques and advanced **Machine Learning (ML)** models, the application processes resumes in formats such as **PDF** and **DOCX**, extracts key features like skills and experience, and assigns them to relevant categories (e.g., "Software Engineer" or "Data Scientist"). This tool minimizes manual effort, enhances accuracy, and optimizes the workflow for HR professionals, enabling them to focus on high-value tasks.

Introduction

Background

Recruitment is often a time-intensive process requiring recruiters to manually review resumes, making it susceptible to human error and inefficiency. In today's fast-paced world, organizations need intelligent automation to streamline this process.

Objectives

- Automate the classification of resumes based on skills, experience, and job roles.

- Improve recruitment efficiency by reducing manual intervention.

- Develop a user-friendly interface for uploading and categorizing resumes.

Scope

The project focuses on:

- Extracting structured data from unstructured resume text.

- Employing **NLP techniques** and **machine learning** models to achieve accurate classification.

- Building an intuitive **web application** for real-world usability.

Related Work

Traditional Methods

Earlier approaches involved rule-based systems or keyword matching, which were rigid and lacked adaptability to diverse resume formats or varying terminologies.

Proposed Approach

Our project leverages advanced **NLP techniques** like **Named Entity Recognition (NER)** and machine learning algorithms, which provide scalability and better accuracy. Additionally, deep learning models like **BERT** (Bidirectional Encoder Representations from Transformers) enable context-aware categorization.

References Articles:

1. Saeed, N. (2024): Building Resume Screening App | Resume Screening with Machine Learning: Job Recommendations, Parsing, & Categorization.

Methodology

Step 1: Dataset Collection and Preprocessing

- Data Sources: Sample resumes were sourced from open datasets and anonymized company archives.

- Preprocessing:

- Converted **PDF/DOCX** files to plain text using libraries like `PyPDF2` and `docx`.

- Cleaned data by removing stopwords, special characters, and irrelevant information.

- Tokenized sentences and words for structured analysis.

Step 2: Feature Extraction

- Extracted the following features:

- Skills: Identified using custom dictionaries and pre-trained NER models.

- Experience: Parsed for job roles and years of experience**.**

- Keywords: Extracted using **TF-IDF** or **embeddings** like Word2Vec**.**

Step 3: Modeling

- Machine Learning Models:

- Algorithms like **Logistic Regression** and **Random Forest** provided baseline classification performance.

- Deep Learning Models:

- Fine-tuned **BERT** and **LSTM** models enhanced accuracy by capturing contextual data.

- Model Comparison:

- Logistic Regression: 80% accuracy

- Random Forest: 85% accuracy

- BERT: 92% accuracy

Step 4: Evaluation

- Metrics Used: Accuracy, Precision, Recall, F1-score.

- Visualization Tools:

- Confusion Matrix to illustrate classification accuracy.

- Feature Importance Graphs to highlight impactful attributes.

Step 5: Deployment

- Built a user-friendly web interface using **Flask** to allow HR professionals to:

  - Upload resumes.

  - View categorized results in real-time**.**

Hardware/Software Requirements

Hardware

- **Minimum:** 8 GB RAM for basic models.

- **Recommended:** GPU for deep learning models like BERT.

Software

- **Programming Language:** Python

- **Libraries:** pandas, numpy, scikit-learn, SpaCy, NLTK, TensorFlow

- **Deployment Tools:** Flask, Streamlit

- **Development Environment:** Jupyter Notebook, VS Code

Experimental Results

Performance

- Traditional Models:

- Logistic Regression: 80% accuracy.

- Random Forest: 85% accuracy.

- Deep Learning Models:

- Fine-tuned BERT: 92% accuracy, with robust performance on varied resume formats.

Visualizations

- Confusion Matrix: Showed improved categorization with deep learning models.

- Graphs: Highlighted the importance of extracted features like “Skills” and “Experience”.

Insights

Deep learning models, particularly **BERT**, significantly outperformed traditional machine learning techniques in terms of both accuracy and adaptability.

Conclusions

The Resume Categorization Application successfully automates the classification process, demonstrating high accuracy and adaptability. By leveraging **state-of-the-art NLP** and **deep learning models,** the tool empowers HR professionals with a scalable and efficient solution for handling large volumes of resumes.

Future Scope

- **Multi-Language Support:** Extend the system to process resumes in multiple languages.

- **Advanced Models: Fine-tune transformers for specific industries or domains.**

- **ATS Integration:** Build an API for seamless integration with **Applicant Tracking Systems.**

GitHub Repository

For more details on the source code and models, check the GitHub repository.

Here are some references you can include for the project:

References:

1. Saeed, N. (2024): Building Resume Screening App | Resume Screening with Machine Learning: Job Recommendations, Parsing, & Categorization.

2. Kaggle. (2024): Resume Datasets - 1

Resume Datasets - 2

3. Kaggle. (2024): Resume Data for Testing

4. Brownlee, J. (2023). Machine Learning Mastery: A Gentle Introduction to Machine Learning*. Machine Learning Mastery.

These references support the methods and tools used in your project for resume categorization and job recommendation systems.