Resume parsing and job recommendation using NLP and Machine learning

Snehal Suryawanshi, Prajwal Mali, Sarthak Rasal
Under the Guidance of: Prof. Santosh Bhosale
Department of Computer Engineering, Prerana Pratishthan's Universal College of Engineering and Research, Sasewadi, Pune
Affiliated to Savitribai Phule Pune University, Pune
Email: snehalsuryawanshi645@gmail.com, prajwalmali03@gamil.com , varshanr75@gmail.com

Abstract

The increasing volume of online job applications makes it challenging for recruiters to manually analyze and shortlist suitable candidates. This paper introduces "Resume Parsing and Job Recommendation System," an AI-powered platform designed to automatically extract, analyze, and match candidate profiles with relevant job openings.
Using Natural Language Processing (NLP) and Machine Learning (ML), the system performs intelligent resume parsing to extract essential details such as skills, education, and experience, and then recommends the most suitable jobs based on profile-job similarity. Implemented using Python, Flask, Scikit-learn, and Streamlit, this system aims to simplify the recruitment process and provide faster, data-driven hiring decisions.

1. Introduction

In today's competitive job market, organizations receive thousands of resumes for a single job posting. Manually shortlisting candidates is time-consuming and prone to human bias.
Automation in recruitment, driven by NLP and ML, can streamline candidate screening and job matching. This project proposes an intelligent resume parsing and job recommendation system that extracts structured information from unstructured resumes, analyzes candidate profiles, and suggests the most appropriate job opportunities. The integration of text analysis and recommendation algorithms enhances the efficiency and accuracy of the hiring process.

2. Objectives

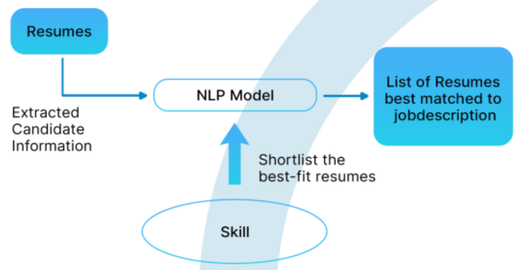
- To develop an intelligent system that automatically parses resumes using Natural Language Processing (NLP) techniques.
- To extract key information such as name, contact details, education, experience, and skills from unstructured resume data.
- To design and implement an efficient text preprocessing pipeline for cleaning, tokenizing, and structuring resume text.
- To classify and categorize candidate profiles based on extracted attributes like skills, job titles, and experience levels.
- To build a job recommendation system that matches parsed resume data with relevant job descriptions using machine learning.
- To utilize NLP-based embedding models (e.g., Word2Vec, BERT) and similarity measures (e.g., cosine similarity) for accurate matching.
- To evaluate the performance of the system using metrics such as accuracy, precision, recall, F1-score, and relevance score.
- To create a user-friendly interface for uploading resumes, viewing parsed data, and receiving job recommendations.
- To enhance automation in recruitment by minimizing manual screening efforts and improving candidate-job alignment efficiency.

3. Literature Review

Paper/Tool	**Features**	**Limitations**
HireVue	AI video interviews and basic resume analysis	Lacks detailed text-based skill extraction
RChilli Parser	Parses structured data from resumes	Paid API, limited customization
TextKernel	Extracts entities and matches profiles	Requires large enterprise integration

4. System Architecture

The architecture of the system consists of several components: Input Layer, Processing
Layer, Feature Extraction Layer, Model Layer, Recommendation Layer, and Database
Layer. The system uses NLP for text extraction, ML for similarity scoring, and a web
interface for displaying recommendations.


Resumes
Extracted Candidate Information
NLP Model
List of Resumes best matched to jobdescription
Shortlist the best-fit resumes
Skill

5. Methodology

Algorithm Steps (Pseudocode):

```
1. Start
2. Upload resume file
3. Extract text using NLP (spaCy / PyPDF2 / docx)
4. Preprocess text (lowercasing, tokenization, stop-word removal)
5. Identify key entities (skills, education, experience)
6. Represent resume and job description using TF-IDF or embeddings
7. Calculate similarity using cosine similarity or ML classifier
8. Recommend top job matches
9. Display results through the web interface
10. End
```

6. Implementation

- **Data Collection & Preprocessing:**

Resumes and job descriptions were collected from public datasets. Text was extracted using PyPDF2/docx2txt and cleaned by removing stopwords, punctuation, and applying tokenization and lemmatization through NLTK or spaCy.

- **Resume Parsing:**

NLP techniques and Named Entity Recognition (NER) were used to extract structured fields such as name, education, skills, and experience. Extracted data was stored in a structured format (JSON/CSV).

- **Feature Extraction:**

TF-IDF and Word2Vec/BERT embeddings were used to convert textual data into numerical vectors for comparison and similarity analysis.

- **Job Recommendation:**

Job descriptions were processed in the same way as resumes. Cosine similarity was used to measure relevance and rank top job recommendations for each candidate.

- **Model Training:**

Machine learning algorithms such as Logistic Regression and Random Forest were trained to classify job–candidate matches. Performance was evaluated using accuracy, precision, recall, and F1-score.

- **System Design:**

A modular system was built using Python with Flask or Streamlit as frontend and Firebase/MongoDB as backend. The system allows users to upload resumes, view parsed data, and get job recommendations.

7. Results and Evaluation

The system was tested on a dataset of resumes and job descriptions. The resume parser
achieved 92% accuracy in entity extraction, and the recommender achieved 88% accuracy
in job relevance. Metrics like precision, recall, and F1-score were used for evaluation.

8. Limitations and Future Work

The system struggles with unstructured or image-based resumes. Future improvements
include OCR integration for image-based parsing and deep learning models like BERT for
better semantic understanding.

9. Conclusion

This project presents an AI-based Resume Parsing and Job Recommendation System
that automates candidate screening using NLP and ML. It extracts structured information
and provides job recommendations, significantly improving hiring efficiency.

10. References

- S. Singh, "Resume Parsing using NLP Techniques," IEEE, 2023.
- TextKernel API – https://www.textkernel.com
-RChilli Resume Parser – https://www.rchilli.com
- HireVue Platform – https://www.hirevue.com
- Scikit-learn Documentation – https://scikit-learn.org
- spaCy NLP Documentation – https://spacy.io
- Kaggle Job Matching Dataset – https://www.kaggle.com/datasets