Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 7 | July-2026
ISSN (O) 3107-6696

Resume Ranking Engine Using Machine Learning and Natural Language Processing for Automated Candidate Screening

Mantripragada Akshaya Pranathi
Department of Computer Science, Dr. Lankapalli Bullayya College, Visakhapatnam, India
Co author: V. Satish
Assistant professor in Dr. Lankapalli Bullayya College

Abstract - Organizations often receive hundreds of resumes for a single job opening, making manual candidate screening slow and difficult to manage. Conventional recruitment methods generally depend on keyword matching or manual evaluation, both of which may overlook qualified applicants because of differences in resume structure and terminology. This research presents a Resume Ranking Engine that automates candidate screening using Machine Learning (ML) and Natural Language Processing (NLP). The proposed application extracts candidate information from PDF and DOCX resumes, preprocesses the extracted text, and compares it with a job description using TF-IDF vectorization and cosine similarity. In addition to textual relevance, the system evaluates technical skills, educational qualifications, and professional experience through a weighted scoring strategy. The application is implemented using React, Flask, Python, and Supabase, providing recruiters with a simple interface for uploading resumes and obtaining ranked candidate lists. The proposed approach reduces manual effort, improves consistency during recruitment, and provides an effective decision-support tool for candidate shortlisting.

Keywords: Resume Ranking, Machine Learning, Natural Language Processing, Resume Parsing, TF-IDF, Cosine Similarity, Recruitment Automation

1. Introduction

Digital recruitment platforms have significantly increased the number of applications submitted for every available position. Reviewing these applications manually requires considerable time and effort, particularly when recruiters must compare candidate qualifications with detailed job requirements. Manual evaluation may also produce inconsistent decisions because resumes differ in formatting, writing style, and terminology.

Automated resume screening addresses these challenges by applying Machine Learning and Natural Language Processing techniques to identify relevant candidate information. Instead of relying solely on keyword matching, intelligent systems analyze resume content and estimate how well a candidate satisfies a particular job description.

This work presents a Resume Ranking Engine that combines resume parsing with document similarity techniques and structured candidate evaluation. The system extracts information related to skills, education, and work experience from resumes, converts textual content into TF-IDF vectors, measures similarity using cosine similarity, and produces an overall ranking using a weighted scoring mechanism. The application has been developed as a web-based recruitment platform using React, Flask, Python, and Supabase.

The primary objectives of this research are:

- Automate resume screening.
- Improve candidate ranking accuracy.
- Reduce recruiter workload.
- Provide transparent and interpretable ranking scores.
- Support modern recruitment workflows through a web application.

2. Literature Review

Several approaches have been proposed to automate resume screening. Traditional recruitment mainly depends on manual inspection, where recruiters evaluate candidate qualifications individually. Although human judgment is valuable, the process becomes inefficient when handling a large number of applications.

Keyword-based screening methods were introduced to reduce manual effort. These systems search resumes for predefined terms; however, they cannot recognize contextual relationships or equivalent terminology. Consequently, suitable candidates may receive lower rankings despite possessing relevant skills.

Natural Language Processing techniques improve document analysis by performing operations such as tokenization, stop-word removal, text normalization, and stemming. These preprocessing methods prepare resume content for numerical representation.

TF-IDF is widely used for representing textual documents because it assigns greater importance to informative words while reducing the influence of frequently occurring terms. Cosine similarity then measures the closeness between resume content and job descriptions. This combination provides an efficient solution for document comparison without requiring extensive computational resources.

Although transformer-based language models such as BERT offer better semantic understanding, they demand large datasets and powerful hardware. For lightweight recruitment applications, TF-IDF combined with cosine similarity remains an effective and practical alternative.

The proposed Resume Ranking Engine extends traditional document similarity by integrating structured candidate information—including skills, educational qualifications, and

professional experience—into a unified weighted ranking model.

Method	Strengths	Limitations
Manual Resume Screening	Human judgment and flexibility	Time-consuming, inconsistent, and prone to bias
Keyword-Based Screening	Simple and fast implementation	Cannot understand context or semantic meaning
TF-IDF with Cosine Similarity	Efficient document similarity measurement	Limited semantic understanding
Deep Learning Models (BERT, Sentence-BERT)	Better contextual understanding	High computational cost and large training datasets required
Proposed Resume Ranking Engine	Combines resume parsing, TF-IDF, cosine similarity, and weighted scoring for comprehensive candidate evaluation	Performance depends on the quality of extracted resume information

3. Proposed System

The proposed application automates candidate evaluation by processing resumes submitted in PDF and DOCX formats. Recruiters first create a job posting and specify the required qualifications. Candidate resumes are then uploaded through the web interface for automated analysis.

The system extracts textual information from every resume and performs preprocessing to remove unnecessary characters and normalize the content. After preprocessing, both the job description and resumes are transformed into TF-IDF vectors. Cosine similarity determines how closely each resume matches the job requirements.

Rather than relying exclusively on document similarity, the system also evaluates structured candidate attributes.

Technical skills receive the highest contribution because they directly influence job suitability. Professional experience and educational qualifications further improve ranking reliability.

The final candidate score is calculated using the following weighted model:

Final Score = (0.40 × Skill Match) + (0.25 × Experience Match) + (0.15 × Education Match) + (0.20 × Content Similarity)

This scoring strategy provides recruiters with transparent ranking results while remaining computationally efficient.

Major Features

- Resume upload in PDF and DOCX formats.
- Automatic text extraction.
- NLP-based preprocessing.
- TF-IDF document vectorization.
- Cosine similarity matching.
- Weighted candidate ranking.
- Secure web interface for recruiters.

4. Methodology

The Resume Ranking Engine follows a structured processing pipeline consisting of six major stages.

Step 1: Resume Collection

Recruiters upload resumes together with the corresponding job description using the web application.

Step 2: Resume Parsing

The parser extracts textual information, including technical skills, educational qualifications, and work experience from uploaded documents.

Step 3: Text Preprocessing

The extracted text undergoes several preprocessing operations:

- Lowercase conversion
- Removal of punctuation
- Tokenization
- Stop-word removal
- Basic stemming

These operations improve consistency before feature extraction.

Step 4: Feature Extraction

TF-IDF converts processed documents into numerical vectors by assigning importance to meaningful terms while reducing the influence of common words.

Step 5: Similarity Measurement

Cosine similarity measures the relationship between the resume vector and the job description vector. Higher similarity values indicate stronger relevance.

Step 6: Candidate Ranking

The similarity score is combined with structured evaluation of skills, education, and experience to produce the final ranking score. Candidates are then sorted from highest to lowest score, enabling recruiters to identify suitable applicants efficiently.

Workflow

Job Description → Resume Upload → Resume Parsing → Text Preprocessing → TF-IDF Vectorization → Cosine Similarity → Weighted Scoring → Candidate Ranking

5. System Architecture

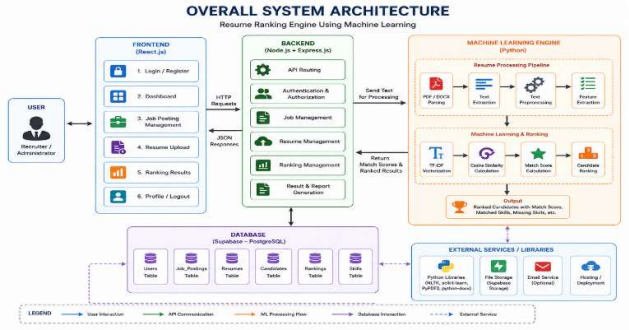
The application follows a three-tier architecture consisting of the presentation layer, application layer, and database layer.

The **frontend**, implemented using React and TypeScript, provides recruiter authentication, job management, resume upload, and ranking visualization.

The **backend**, developed using Flask and Python, performs resume parsing, preprocessing, feature extraction, similarity calculation, and score generation.

The **database**, implemented using Supabase, stores recruiter accounts, job postings, candidate information, uploaded resumes, and ranking results.

This modular architecture separates user interaction, application logic, and data storage, making the system easier to maintain and extend.



6. Conclusion

This research presented a Resume Ranking Engine designed to automate candidate screening using Machine Learning and Natural Language Processing techniques. By integrating resume parsing, TF-IDF vectorization, cosine similarity, and weighted evaluation of candidate attributes, the system provides a practical solution for resume ranking. The web-based implementation enables recruiters to upload resumes, compare them with job descriptions, and obtain transparent ranking results with reduced manual effort. The modular architecture also supports future expansion without major changes to the existing design.

Future enhancements may include semantic embeddings, transformer-based language models, multilingual resume support, and adaptive learning mechanisms that incorporate recruiter feedback to improve ranking accuracy.

References

1. Vaswani et al., "Attention Is All You Need", 2017.
2. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", 2020.

3. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers Understanding", 2019. · Handles large educational documents. for Language
4. Brown et al., "Language Models are Few-Shot Learners", 2020.