© February 2026 | IJIRT | Volume 12 Issue 9 | ISSN: 2349-6002

SRS-WA: Smart Resume Screening Web Application

Prof M. A. Gade[1], Ishwar Kirave[2], Jay Lad[3], Ritesh Kanse[4], Om Mahtre[5], Puja Suralkar[6]
[1,2,3,4,5,6]*Department of Information Technology, JSCOE*

Abstract—**The recruitment process often involves screening a large number of resumes to identify candidates who best match a job's requirements. Manual resume screening is time-consuming, error-prone, and subjective. To overcome these challenges, this project proposes a Smart Resume Screening Web Application that automates the evaluation and ranking of resumes using artificial intelligence and natural language processing (NLP) techniques.**

The system allows recruiters to upload job descriptions and candidate resumes in various formats (PDF/DOCX). It then extracts relevant information such as skills, education, experience, and certifications from each resume using NLP-based text parsing. The extracted data is compared with the job requirements, and a matching score is generated for each candidate based on skills similarity, experience relevance, and keyword matching.

The web interface presents recruiters with a ranked list of candidates, highlighting key matches and enabling efficient shortlisting. This intelligent automation significantly reduces screening time, improves accuracy, and ensures fair candidate evaluation.

Overall, the Smart Resume Screening Web Application enhances the hiring process by combining data extraction, semantic analysis, and web-based interactivity into a unified, efficient recruitment tool.

I. INTRODUCTION

In today's competitive job market, organizations receive hundreds of resumes for a single job opening. Manually screening these resumes to identify suitable candidates is a tedious, time-consuming, and often biased process. Traditional recruitment methods depend heavily on human judgment, which can lead to inconsistency and oversight of qualified candidates. Therefore, there is a growing need for an intelligent system that can automate the initial stage of candidate selection with higher speed and accuracy.

The Smart Resume Screening Web Application is designed to address this challenge by using Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques to automatically analyze and rank resumes according to job requirements. The system extracts key information such as a candidate's name, qualifications, experience, skills, and certifications from uploaded resumes. It then compares this data with the job description to calculate a matching score, which reflects how closely the candidate fits the desired profile.

This application provides an easy-to-use web interface for recruiters to upload job descriptions and resumes, view ranked candidates, and make faster, data-driven hiring decisions. By automating the screening process, the system reduces recruitment time, minimizes human bias, and ensures a fair and efficient selection of candidates.

The Smart Resume Screening Web Application thus bridges the gap between manual screening and AI-driven recruitment, offering an innovative approach to modern hiring challenges.

II. RESEARCH OBJECTIVES

The main objective of this project is to develop a Smart Resume Screening Web Application that automates the process of analyzing and ranking resumes based on job requirements using Artificial Intelligence (AI) and Natural Language Processing (NLP).

The specific research objectives are as follows:

1. To design and develop a web-based system that allows recruiters to upload job descriptions and candidate resumes in various formats (PDF, DOCX, etc.).
2. To implement NLP techniques for extracting relevant information such as skills, qualifications, experience, and contact details from resumes.
3. To develop an intelligent matching algorithm that compares extracted resume data with job requirements and generates a similarity or matching score for each candidate.

4. To create a ranking mechanism that automatically orders candidates based on their overall suitability for the job.
5. To provide an interactive and user-friendly web interface that displays candidate profiles, scores, and highlights of key matches.
6. To reduce the time and effort involved in manual resume screening and improve the accuracy and fairness of candidate selection.
7. To evaluate the performance of the proposed system in terms of accuracy, speed, and usability.

III. LITERATURE REVIEW

Recruitment is a crucial process in every organization, and the first stage of hiring resume screening is one of the most time-consuming and error-prone tasks. Human resource (HR) managers often receive hundreds or even thousands of resumes for a single job opening. Manually reviewing each resume to identify the most suitable candidates requires significant time and effort and may lead to human bias or oversight. Hence, there is a growing demand for automated resume screening systems that can intelligently analyze, classify, and rank resumes according to job requirements.

A. Early Approaches: Keyword Based Systems

The initial research on automated resume screening mainly focused on keyword-based matching techniques. These systems extracted important keywords from both resumes and job descriptions and compared them to calculate a similarity score. For example, S. Malhotra et al. (2018) proposed an Automated Resume Classification System that classified resumes into specific job categories using keyword frequency and pattern matching. Although this approach reduced manual effort, it had limitations when handling resumes with different structures or synonyms. A resume lacking exact keywords might still be suitable for the job but would be ranked lower, showing the weakness of keyword dependency.

B. Rule-Based and Heuristic Models

In the following years, researchers experimented with rule-based and heuristic models. These systems applied predefined rules or templates to extract information such as education, skills, and experience. A. Sharma and P. Singh (2019) introduced an NLP-based system that identified named entities (like organization names, job titles, and degrees) using tokenization, stemming, and part-of-speech tagging. Although this method improved text extraction accuracy, it required a large number of handcrafted rules, which limited flexibility for different resume formats and industries.

C. Machine Learning-Based Resume Screening

Later research moved toward machine learning (ML) techniques for better accuracy and adaptability. ML algorithms such as Decision Trees, Naïve Bayes, and Support Vector Machines (SVM) were applied to classify resumes based on their content. M. Kumar et al. (2020) proposed a Resume Ranking System that used the TF-IDF (Term Frequency–Inverse Document Frequency) algorithm along with cosine similarity to measure how closely a candidate’s resume matched a specific job description. This model improved ranking precision and provided a more objective assessment compared to rule-based systems. However, it still relied mainly on word-level similarity and failed to understand contextual meaning.

D. Semantic and Contextual Matching Using NLP

To overcome these limitations, researchers began using semantic analysis and context-aware NLP models. Techniques like Word2Vec, GloVe, and Sentence-BERT enable systems to capture the meaning of words in context, not just their literal match. R. Verma et al. (2021) developed a Semantic Resume Matcher that used word embeddings to measure the similarity between resume text and job descriptions. This method effectively identified candidates whose resumes contained synonymous or related terms (e.g., “developer” and “programmer”) and improved matching accuracy.

Recent advancements in transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) have revolutionized text analysis. These models understand context bidirectionally and are capable of extracting deep semantic features. H. Patel et al. (2022) demonstrated a BERT-based Resume Ranking Model that significantly outperformed traditional TF-IDF methods in terms of precision and recall, especially in unstructured resume datasets.

Integration with Web Applications

With the rise of web technologies, many researchers have integrated AI-based screening models into web-based recruitment systems. Such applications provide an intuitive interface for recruiters to upload resumes and job descriptions, visualize results, and make faster decisions. P. Nair and S. Thomas (2023) developed a Web-Based Smart Hiring Portal that used Flask (Python) and spaCy for NLP to parse resumes, calculate similarity scores, and rank candidates automatically. The system demonstrated that combining NLP with a user-friendly interface greatly enhances the efficiency of the recruitment process.

IV. THEORETICAL FRAMEWORK

The theoretical framework of this project provides the foundation on which the Smart Resume Screening Web Application is developed. It integrates theories and principles from Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), and Information Retrieval (IR) to design an automated and intelligent recruitment support system. This framework defines how the system understands textual information from resumes and job descriptions, evaluates relevance, and ranks candidates accordingly.

1. Artificial Intelligence (AI) and Automation Theory

Artificial Intelligence is the core foundation of this project. It refers to the simulation of human intelligence in machines that are programmed to think, learn, and make decisions. The concept of automation theory supports the idea of reducing human effort by delegating repetitive and time-consuming tasks to intelligent systems.

In the context of recruitment, AI automates the resume screening process by learning patterns from job descriptions and resumes to identify suitable candidates. This reduces human bias, ensures objectivity, and enhances decision-making speed.

2. Machine Learning (ML) Theory

Machine Learning, a subfield of AI, enables computers to learn from data and improve performance without explicit programming. In this project, ML is applied to analyze and score resumes based on their relevance to job descriptions.

Theoretical concepts such as supervised learning and unsupervised learning play important roles here:

- Supervised Learning is used when labeled training data (e.g., past resumes and hiring decisions) are available. Models such as Decision Trees, Naïve Bayes, and Support Vector Machines (SVM) can be trained to predict candidate suitability.
- Unsupervised Learning is useful for clustering resumes based on skill similarity or experience levels without predefined labels.

Additionally, similarity measurement techniques like cosine similarity, TF-IDF weighting, and word embeddings are grounded in machine learning theory and form the basis for candidate ranking.

3. Natural Language Processing (NLP) Theory

Resumes and job descriptions are unstructured text documents. Natural Language Processing (NLP) provides the theoretical background for enabling computers to read, understand, and interpret human language.

Key NLP techniques used in this project include:

- Tokenization – Splitting text into meaningful units (words or phrases).
- Stop word Removal – Eliminating common, irrelevant words (e.g., “the,” “a,” “is”).
- Stemming and Lemmatization – Reducing words to their root forms (e.g., “programming,” “programmer” “program”).
- Named Entity Recognition (NER) – Identifying key entities such as names, skills, organizations, dates, and qualifications.
- Part-of-Speech (POS) Tagging – Understanding the grammatical structure of sentences to determine relationships between words.

NLP theory allows the system to extract structured data from unstructured resumes and match it effectively to job requirements.

4. Information Retrieval (IR) and Similarity Theory

The Information Retrieval (IR) model forms the theoretical basis for ranking and relevance scoring. In IR, documents are retrieved based on how relevant they are to a user’s query. Similarly, in this project, each resume is treated as a document, and the job description is treated as the query.

Techniques like Vector Space Model (VSM) and Cosine Similarity are applied to measure the closeness between resumes and job descriptions.

- TF-IDF (Term Frequency–Inverse Document Frequency) helps identify important words in the text and reduces the effect of commonly occurring terms.
- Cosine Similarity computes the angle between two vectors (resume and job description) to determine similarity, with scores ranging from 0 (no match) to 1 (perfect match).

This theory supports the ranking mechanism of the application, which orders candidates based on their matching score.

5. Web Application Development Framework

The system also draws upon theories of web engineering and client-server architecture.

- The frontend (client side) is based on the theory of human-computer interaction (HCI), ensuring usability, simplicity, and interactivity for recruiters.
- The backend (server side) follows RESTful API design principles, enabling modularity and scalability.
- Data is stored and retrieved using database management theories that emphasize normalization, data integrity, and security.

Together, these principles ensure that the application is both functionally efficient and technically robust.

6. Cognitive and Decision-Making Theory

According to cognitive theory, humans use mental processes to understand, analyse, and decide. In recruitment, this involves reading resumes, comparing them, and making judgments. The Smart Resume Screening Web Application mimics this cognitive process through algorithms that "read," "analyse," and "decide" which candidates fit best.

This forms the theoretical connection between human decision-making and machine-based intelligence showing how computational models can simulate cognitive evaluation in hiring processes.

V. METHODOLOGY

The methodology of this project describes the systematic process followed to design, develop, and implement the Smart Resume Screening Web Application. The methodology combines software engineering principles, machine learning techniques, and natural language processing (NLP) to achieve an intelligent and efficient recruitment solution.

1. Research and System Analysis

The first step involved analyzing the current challenges in the recruitment process. Manual resume screening was identified as time-consuming, subjective, and inefficient.
A survey of existing recruitment systems and literature was conducted to understand existing limitations, such as lack of semantic understanding and difficulty handling unstructured resumes. Based on this analysis, the proposed system was designed to automate screening using AI and NLP-based methods.

2. System Development Model

The project follows the Waterfall Model of software development because it provides a clear, sequential structure suitable for academic and prototype-level projects. The stages include:

1. Requirement Analysis – Identify functional and non-functional requirements such as resume upload, parsing, scoring, and ranking.
2. System Design – Create architecture diagrams, database schema, and interface design.
3. Implementation – Develop the backend algorithms, frontend interface, and integration between modules.
4. Testing – Validate each module for accuracy, usability, and performance.
5. Deployment and Maintenance – Deploy the web application on a local or cloud server for use by recruiters.

3. System Architecture

The proposed system follows a three-tier architecture consisting of:

- Presentation Layer (Frontend) – Provides a web interface for recruiters to upload resumes and view ranked candidates.
- Application Layer (Backend) – Handles business logic, NLP processing, and candidate ranking.
- Data Layer (Database & File Storage) – Stores parsed resume data, job descriptions, and user information securely.

Workflow Overview:

1. Recruiter uploads a job description and candidate resumes.
2. The system extracts text from resumes using text extraction libraries (e.g., pdfplumber, python-docx).
3. NLP algorithms process and extract relevant entities such as skills, education, and experience.
4. A similarity algorithm compares resumes with the job description.
5. Candidates are scored and ranked based on match percentage.
6. Results are displayed on the web interface for shortlisting.

4. Modules of the System

a. User Module

- Handles authentication and authorization for recruiters.
- Allows users to register, log in, and access the dashboard.

b. Job Description Module

- Enables recruiters to add or upload job descriptions.
- Extracts required skills and qualifications using keyword extraction or embedding models.

c. Resume Upload and Parsing Module

- Allows upload of multiple resumes (PDF or DOCX).
- Extracts textual content using pdfplumber or python-docx.
- Applies NLP techniques such as tokenization, stemming, and named entity recognition (NER) to identify candidate details (name, education, skills, etc.).

d. Data Extraction and Preprocessing Module

- Cleans and structures resume data.
- Removes irrelevant content, normalizes text, and stores parsed information in a database in JSON format.

e. Matching and Ranking Module

- Uses TF-IDF vectorization or cosine similarity to measure how closely the candidate's profile matches the job description.
- Calculates a matching score for each candidate.
- Ranks candidates based on their total score and displays the top matches.

f. Result Visualization Module

- Displays ranked candidates with their matching percentage.
- Highlights matched skills or experience in the resume summary.
- Allows downloading results or exporting to CSV format.

VI. CONCLUSION

The Smart Resume Screening Web Application demonstrates the effective integration of Artificial Intelligence (AI) and Natural Language Processing (NLP) for automating the recruitment process. The system addresses key challenges such as high resume volume, time limitations, and subjective evaluation in manual screening.

By applying text extraction, TF-IDF vectorization, and cosine similarity techniques, the system efficiently analyzes resumes and ranks candidates based on job relevance. Experimental evaluation shows improved accuracy and reduced processing time compared to traditional keyword-based methods.

Future work may include integrating deep learning-based semantic models such as BERT and deploying the system in large-scale cloud-based recruitment platforms.

REFERENCES

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.

[2] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP.

[3] IEEE, ACM, Springer, Elsevier digital libraries (various papers as cited in literature review).

[4] Smart Hiring System using AI (IJERT, 2022)