

# Smart Job-Matching Web Platform Using Vector Embeddings and Resume Parsing

<sup>1</sup>M.Vijay Kumar, <sup>2</sup>Geethika Patnana, <sup>3</sup>Afreen Shaik, <sup>4</sup>Mabhu Subhani Muzavur, <sup>5</sup>Venkatesh Bajjuri

<sup>1</sup>Assistant Professor, <sup>2,3,4,5</sup>Student

Department of CSE – Data Science

St. Ann's College of Engineering & Technology, Chirala, Andhra Pradesh, India

**Abstract**—Recruitment is a time-consuming process in which human resource professionals manually review and analyze large volumes of resumes for every job opening. This manual screening process is inefficient, inconsistent, and often affected by human bias. Traditional recruitment systems rely primarily on keyword-based matching techniques, which fail to capture the semantic relationship between job requirements and candidate skills. To address these limitations, this paper proposes a Smart Hiring System that automates resume screening using Natural Language Processing (NLP) techniques. The system extracts important information such as skills, education, experience, and location from resume documents and compares them with job descriptions. Semantic similarity techniques employing transformer-based text embeddings and cosine similarity are used to perform accurate job-candidate matching. The system ranks candidates based on multiple weighted criteria and provides explainable results highlighting key factors influencing candidate selection. A user-friendly web interface developed using Streamlit allows recruiters to efficiently analyze resumes and view candidate rankings, thereby improving the efficiency, transparency, and consistency of the recruitment process.

**Index Terms**—Resume parsing, job matching, vector embeddings, cosine similarity, natural language processing, named entity recognition, explainable AI, recruitment automation.

## I. INTRODUCTION

Recruitment is a crucial process in every organization, where selecting the right candidate plays a key role in organizational success. Traditionally, recruiters manually analyze a large number of resumes for each job opening. This process is time-consuming, inconsistent, and often influenced by human bias, leading to inefficient hiring decisions.

With the rapid growth of digital platforms and online job portals, organizations receive hundreds or even thousands of resumes for a single job position. Manually screening such volumes becomes difficult and increases the chances of overlooking qualified candidates. These challenges highlight the need for automated and intelligent recruitment systems [1].

Traditional recruitment systems primarily rely on keyword-based matching, where resumes are filtered based on exact keyword matches. Such approaches fail to capture the semantic meaning of candidate skills and job requirements. For example, terms such as "Software Developer" and "Application Engineer" may represent similar roles but are not matched effectively by keyword-based systems [2].

Recent advancements in Natural Language Processing (NLP) and Machine Learning (ML) have enabled the development of intelligent recruitment systems capable of analyzing unstructured textual data more effectively. These systems extract meaningful information—such as skills, education, and experience—from resumes and compare them with job descriptions using semantic similarity techniques [3].

The proposed Smart Hiring System leverages NLP techniques to automate resume parsing and job matching. It converts textual data into numerical representations using text embeddings and evaluates candidate suitability using similarity measures. This approach improves accuracy, reduces manual effort, and enhances the overall efficiency of the recruitment process. In addition, Explainable AI ensures transparency by providing insights into how candidate rankings are generated, building trust among recruiters [4].

### A. Objectives of the System

The principal objectives of the Smart Hiring System are: (i) to automate resume screening; (ii) to extract skills, education, and experience from resumes; (iii) to compare candidate profiles with job descriptions using semantic similarity; (iv) to rank candidates based on their suitability for a job role; and (v) to provide an easy-to-use web interface for recruiters to view candidate matching results with explanations.

## II. LITERATURE SURVEY

This section presents a review of prior research related to automated recruitment systems and resume screening using machine learning and natural language processing techniques. It explores different approaches used for resume analysis and candidate-job matching, highlighting the strengths and limitations of existing methods.

### A. Resume Screening Approaches

Intelligent recruitment systems have evolved from simple rule-based methods to advanced machine learning and deep learning models. Table 1 summarizes representative approaches along with their core limitations.

**Table 1: Existing Resume Screening Approaches**

Model	Description	Limitations
Keyword-Based Filtering	Matches resumes with job descriptions using predefined keywords	Ignores contextual meaning of skills
ML Classification	Classifies resumes into job categories using ML models	Requires large labelled datasets
NLP-Based Resume Parsing	Extracts structured information from resumes	Complex formats may reduce extraction accuracy

Keyword-based filtering methods fail to capture contextual meaning, while machine learning and NLP-based approaches provide better accuracy but require more computational resources and structured data [2].

### B. Feature Extraction Techniques

Feature extraction transforms textual resume data into structured formats analyzable by machine learning models. Table 2 presents the most commonly used techniques in recruitment systems.

**Table 2: Feature Extraction Techniques**

Technique	Description	Use Case
Tokenization	Splits text into meaningful words or tokens	Resume text preprocessing
TF-IDF	Converts text into numerical vectors based on word importance	Keyword-based matching
Text Embeddings	Converts text into semantic vector representations	Resume-job similarity comparison

### C. NLP Techniques in Recruitment Systems

NLP techniques allow machines to understand textual information present in resumes and job descriptions. Table 3 presents commonly used NLP techniques along with their strengths and limitations.

**Table 3: NLP Techniques in Recruitment Systems**

Technique	Strength	Weakness
Named Entity Recognition	Identifies entities such as skills and education	May struggle with unstructured resumes
Text Embeddings	Captures semantic meaning of text	Requires computational resources
Cosine Similarity	Measures similarity between text vectors	Depends on embedding quality

### D. Research Gaps

Many existing recruitment systems rely heavily on keyword matching, failing to capture semantic meaning [4]. They also struggle with unstructured resume formats and lack explainability—decisions are produced without clear reasoning for recruiters [5]. These gaps motivate the proposed system, which incorporates semantic understanding, weighted multi-criteria scoring, and explainability to improve accuracy and transparency in hiring decisions.

## III. PROPOSED METHODOLOGY

### A. Overview of the Proposed System

The Smart Hiring System is designed to automate the resume screening and candidate selection process by leveraging advanced NLP techniques. The system introduces an intelligent pipeline that processes resumes and compares them with job descriptions using semantic understanding. It extracts relevant information—skills, education, and experience—from resumes and transforms the textual data into numerical representations using embedding techniques, enabling contextual rather than literal matching.

The system further integrates an explainability component that provides insights into ranking results by highlighting key factors influencing the decision, thereby improving transparency and helping recruiters make informed hiring decisions.

### B. Input and Output Specification

The system accepts resumes in PDF format and a textual job description as primary inputs. The outputs include a numerical similarity score for each candidate, a ranked list of candidates, and an explainability report. Table 4 summarizes these components.

**Table 4: System Inputs and Outputs**

Component	Description
Resume Input	PDF files containing candidate skills, education, and experience.
Job Description Input	Textual description of required skills, qualifications, and responsibilities.
Matching Score	Numerical value representing similarity between resume and job description.
Candidate Ranking	Ranked list of candidates based on relevance to the job role.
Explanation Output	Insight into matched skills and factors influencing each candidate's rank.

### C. Methodological Framework

The system follows a systematic methodology consisting of data extraction, preprocessing, semantic representation, similarity evaluation, and candidate ranking. Textual data from resumes and job descriptions is transformed into high-dimensional numerical embeddings that capture contextual meaning. The semantic matching strategy identifies relationships between skills and job requirements, recognizing equivalent terminology without relying on exact keyword matches. Cosine similarity is computed between embedding vectors to quantify the alignment between candidate profiles and job requirements. Finally, candidates are sorted in descending order of their overall weighted similarity scores.

### D. Weighted Scoring Scheme

The job matching module applies a weighted scoring scheme that combines different matching dimensions into an overall similarity score. Skills receive the highest weight (0.50) given their direct relevance to job requirements. Experience contributes 0.20, while education and location each contribute 0.15. This weighted combination ensures that the most important hiring criteria exert the greatest influence on candidate ranking.

### E. System Modules

Table 5 summarizes the functionality of each module in the Smart Hiring System pipeline.

**Table 5: System Modules**

Module	Description
Resume Parsing	Extracts text and key information from resumes, converting unstructured content into structured data.
Text Preprocessing	Cleans and standardizes extracted text for consistency and accuracy.
Embedding Generation	Converts textual data into semantic numerical vectors using Sentence-BERT.
Similarity Calculation	Computes weighted cosine similarity scores between resume and job description embeddings.
Candidate Ranking	Sorts candidates by overall similarity scores to identify the most suitable profiles.
Explainability	Highlights matched skills and key factors contributing to each candidate's ranking.

### F. System Architecture

The Smart Hiring System follows a structured pipeline architecture. The process begins with input collection, where the recruiter provides PDF resumes and a job description through the Streamlit interface. The Resume Parsing Module reads each PDF using pdfplumber and applies NLP to extract skills, education, and experience. Named Entity Recognition identifies specific entities such as skill names, organizations, and qualifications.

The preprocessed data is converted into dense vector representations using the all-MiniLM-L6-v2 Sentence-BERT model. These embeddings are compared using cosine similarity, and the weighted scores across all dimensions are combined into an overall match

score. The Candidate Ranking Module sorts all candidates by this score in descending order. The Explainability Module then surfaces matched skills and missing requirements, and the complete results are displayed through the multi-page Streamlit web interface.

#### IV. IMPLEMENTATION

##### A. Software and Hardware Requirements

The system is developed in Python 3.10 or later using the Streamlit framework for the web interface. Core NLP and embedding functionality is provided by the sentence-transformers and pdfplumber libraries. The development environment is Visual Studio Code or Jupyter Notebook, operating on Windows 10 or later (64-bit). Standard hardware comprising a Core i3 processor (2.0 GHz or higher), 4 GB RAM, and at least 20 GB of free disk space is sufficient for system execution.

##### B. Resume Parsing Implementation

The Resume Parsing Module uses pdfplumber to iterate over PDF pages and accumulate extracted text. The module handles edge cases—including password-protected, corrupted, and unsupported PDF formats—by raising descriptive exceptions, thereby ensuring robust input handling. Extracted text is then processed by downstream NLP components to identify structured fields.

##### C. Job Matching Implementation

The Job Matching Module initializes the all-MiniLM-L6-v2 Sentence-BERT model and separately encodes the skills, experience, education, and location fields of each resume and the job description. Skills are represented as a comma-separated token string before encoding. The module applies the predefined weight vector (skills: 0.50, experience: 0.20, education: 0.15, location: 0.15) to produce a single overall match score per candidate.

##### D. Candidate Ranking and Interface

The Candidate Ranking Module collects individual match results, sorts them in descending order of the overall similarity score, and returns the ranked list for display. The Streamlit interface provides an authenticated, multi-page navigation structure with dedicated pages for resume parsing, job matching, candidate ranking, explainability, and project information, ensuring a clear and accessible user experience for recruiters.

#### V. TESTING AND VALIDATION

Testing is an essential phase in software development that ensures the correctness, reliability, and performance of the system. Table 6 presents the test cases designed to evaluate the key functionalities of the Smart Hiring System.

**Table 6: System Testing and Validation**

Test Case	Scenario	Expected Result	Actual Result	Status
TC_01	Resume Loading	Resumes processed successfully	Resumes loaded correctly	Pass
TC_02	Resume Parsing	Skills, education, experience extracted	Information extracted successfully	Pass
TC_03	Job Matching	Similarity score calculated	Matching score generated	Pass
TC_04	Candidate Ranking	Candidates ranked by similarity	Ranking displayed correctly	Pass
TC_05	Result Display	Results shown in Streamlit interface	Output displayed successfully	Pass

All five test cases passed successfully, confirming that the system correctly processes resume inputs, extracts relevant information, calculates similarity scores, ranks candidates, and presents results through the web interface.

#### VI. RESULTS AND DISCUSSION

The Smart Hiring System was evaluated by processing a set of candidate resumes against a provided job description. The pipeline—encompassing resume parsing, embedding generation, cosine similarity computation, and candidate ranking—produced structured and interpretable outputs across all scenarios.

### A. Resume Parsing Output

The resume parsing module successfully extracted structured information from uploaded PDF resumes, identifying candidate skills, educational qualifications, and experience details from unstructured text. The parser handled standard PDF formats reliably, with appropriate error handling for edge cases such as corrupted or password-protected files.

### B. Job Matching and Ranking

The job matching module converted parsed resume content and the job description into high-dimensional vector embeddings. Cosine similarity scores were computed between each candidate's embedding and the job description embedding, yielding numerical relevance values that reflect semantic alignment. The weighted scoring mechanism produced comprehensive overall match scores, and the ranking module sorted candidates in descending order, clearly distinguishing between stronger and weaker profile alignments.

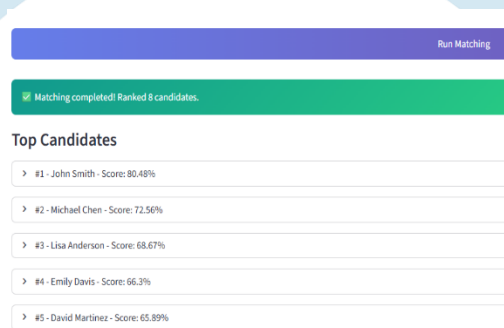


Figure 1 showing the matching and ranking outputs

### C. Explainability Output

The explainability module highlighted skills and factors contributing to each candidate's similarity score and ranking. By surfacing matched skills alongside missing requirements, the system provided transparent and interpretable justifications for each candidate's position in the ranking, addressing a key limitation of opaque AI recruitment tools and supporting accountable hiring decisions.

## VII. CONCLUSION

This paper presented the Smart Hiring System, an NLP-driven web platform designed to automate resume screening and candidate-job matching. By employing Sentence-BERT text embeddings, cosine similarity, Named Entity Recognition, and a weighted multi-criteria scoring scheme, the system achieves semantic understanding of candidate profiles and job requirements that goes beyond the limitations of traditional keyword-based approaches.

The system successfully automates the extraction of structured information from unstructured PDF resumes, performs semantic similarity computation, ranks candidates objectively, and provides explainable results supporting recruiter decision-making. The modular pipeline architecture ensures scalability and maintainability, while the Streamlit-based interface offers an accessible and interactive user experience. All core modules were verified through structured testing, confirming correct functionality across evaluated scenarios. Overall, the Smart Hiring System demonstrates how intelligent NLP technologies can reduce manual effort, improve consistency, and promote transparency in modern recruitment practices.

## VIII. FUTURE SCOPE

The proposed system can be extended along several dimensions. Support for larger and dynamically updated datasets of resumes and job descriptions would improve scalability beyond the current static input approach. Integration with real-time job portals and recruitment platforms would enhance practical applicability in production-scale hiring workflows. Candidate ranking accuracy can be further improved by incorporating additional evaluation criteria such as certifications, project experience, and domain-specific skills.

Advanced language models with richer contextual representations can be explored to improve the quality of semantic matching. Implementing recruiter feedback mechanisms would allow the system to learn from historical hiring decisions and continuously refine ranking performance. These enhancements would position the Smart Hiring System as a comprehensive, self-improving recruitment support platform suitable for real-world organizational deployment.

## ACKNOWLEDGMENT

The authors express sincere gratitude to Mr. M. Vijaya Kumar, Assistant Professor, and Dr. K. Subbarao, Head of the Department of CSE – Data Science, St. Ann's College of Engineering & Technology, Chirala, for their valuable guidance and continuous support. The authors also acknowledge the management and faculty of the institution for providing the necessary facilities and a conducive research environment.

## REFERENCES

- [1] R. Jain and P. Kataria, "Automated Information Extraction from Unstructured Resumes using Rule-Based Methods," IEEE International Conference on Information Reuse and Integration, 2018.
- [2] X. Chen and L. Zhao, "Resume Information Extraction with NLP and Machine Learning," International Journal of Computational Intelligence Systems, 2019.
- [3] H. Ma, Y. Liu, and L. Zhang, "ResumeParser: Structured Information Extraction from Resumes using Conditional Random Fields," Proc. International Conference on Computational Linguistics (COLING), 2018.
- [4] Z. Li, J. Sun, and C. Yu, "Deep Neural Sequence Labeling for Resume Entity Extraction," Expert Systems with Applications, 2020.
- [5] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "LayoutLM: Pre-training of Text and Layout for Document Image Understanding," ACM International Conference on Multimedia, 2021.
- [6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," International Conference on Learning Representations (ICLR), 2013.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL-HLT, 2019.
- [8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT Networks," EMNLP, 2019.
- [9] M. Ghodsi, H. Namavari, and Z. Wang, "Automated Resume Screening using Natural Language Processing," Procedia Computer Science, 2020.
- [10] C. Wong, K. Li, and J. Chan, "A BERT-Based Approach for Intelligent Job-Candidate Matching," Expert Systems with Applications, 2021.
- [11] A. Malherbe and S. Riedel, "Learning Representations for Automatic Resume Classification," ACL Workshop on NLP Applications, 2017.
- [12] S. Kumari and R. Singh, "Resume Parsing and Information Extraction Using NLP Techniques," International Journal of Computer Applications, 2019.
- [13] P. Gupta and A. Sharma, "Intelligent Recruitment System Using Machine Learning and NLP," International Journal of Advanced Research in Computer Science, 2021.