Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

NexusAi: AI-Based Automated Resume Screening and Job Matching System Using NLP

[1]Anjali Chandra,[2]Awaish Ahmed, [3]Kanha Verma, [4]Ishan Jain
[1,2,3,4]Department of Artificial Intelligence and Machine Learning
Shri Shankaracharya Institute of Professional Management and Technology
Raipur, Chhattisgarh, India
[1]anjali.chandra68@gmail.com, [2]aqureshi@ssipmt.com, [3]kanhaVerma2023@gmail.com,
[4]ishan@ssipmt.com

Abstract—The recruitment process has become increasingly complex due to the large volume of job applications received by organizations. Traditional Applicant Tracking Systems (ATS) primarily rely on keyword-based filtering, which often fails to capture the semantic meaning and contextual relevance of candidate profiles. This paper proposes an AI-Based Automated Resume Screening and Job Matching System using Natural Lan-guage Processing (NLP) and Machine Learning techniques. The system automatically parses resumes, performs text preprocess-ing, extracts key features using Term Frequency–Inverse Docu-ment Frequency (TF-IDF) vectorization, and computes similarity between candidate profiles and job requirements using cosine similarity. The system generates a match percentage score along with matched and missing skill insights, enabling faster, fairer, and more consistent hiring decisions. A user-friendly interface is implemented using Streamlit, and the system is integrated with the Google Gemini API for enhanced semantic analysis. The proposed approach significantly reduces manual effort, minimizes human bias, and improves recruitment efficiency and accuracy.

Keywords—Automated Resume Screening, Applicant Track-ing System, Natural Language Processing, TF-IDF, Cosine Sim-ilarity, Machine Learning, Job Matching, Recruitment Automa-tion, Streamlit, Gemini API

I. INTRODUCTION

In today's fast-paced digital world, the recruitment process has become increasingly complex due to the large volume of job applications received by organizations. Human Resource (HR) departments often spend significant time and effort manually screening resumes to identify suitable candidates. This traditional approach is not only time-consuming but also prone to human bias, inconsistency, and errors, which can affect the overall quality of hiring decisions.

With the rapid advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP), there is a growing opportunity to automate and enhance the recruitment process. AI-driven systems can efficiently analyze large datasets, ex-tract meaningful information, and make intelligent decisions based on predefined criteria [1]. In the context of recruitment, these technologies can be used to develop automated systems that screen resumes, extract relevant skills, and match candi-dates with job descriptions more accurately.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

An Applicant Tracking System (ATS) is widely used by organizations to manage recruitment workflows. However, traditional ATS platforms primarily rely on keyword-based filtering, which often fails to capture the semantic meaning and

contextual relevance of candidate profiles [2]. This limitation can result in the rejection of qualified candidates or the selection of less suitable ones.

To address these challenges, this paper proposes an Auto-mated Resume Screening and Job Matching System using NLP and Machine Learning techniques. The system is designed to intelligently parse resumes, perform text preprocessing, extract key features, and compute similarity between candidate profiles and job requirements. By leveraging techniques such as tokenization, stop-word removal, TF-IDF vectorization, and cosine similarity, the system can evaluate and rank candidates based on their relevance to a given job role.

The primary objective of this project is to improve the effi-ciency, accuracy, and fairness of the recruitment process. By automating resume screening and job matching, the proposed system reduces manual effort, minimizes bias, and enables faster decision-making. Additionally, the system is scalable and can be integrated into real-world recruitment platforms to handle large volumes of applications effectively.

A. Objectives

The primary objectives of this project are:

•To automate the resume screening process by automati-cally extracting relevant information such as skills, qual-ifications, and work experience from unstructured text documents.

•To implement effective NLP techniques for understanding unstructured textual data, including tokenization, stop-word removal, lemmatization, and text normalization.

•To enhance the accuracy of candidate-job matching using TF-IDF feature extraction and cosine similarity measures.

•To rank candidates based on their suitability for a given job role by assigning match scores.

•To minimize human bias in the recruitment process by providing a consistent, data-driven evaluation mechanism.

•To design a user-friendly interface that allows recruiters to upload resumes, input job descriptions, and view results in an organized manner.

B. Problem Statement

The recruitment process involves handling a large number of resumes, making manual screening time-consuming, inef-ficient, and prone to human bias. Traditional ATS platforms

primarily rely on keyword-based filtering, which often fails to capture the contextual meaning of candidate skills and job requirements. This can lead to inaccurate candidate-job matching and missed opportunities for qualified applicants. Therefore, there is a need for an intelligent

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

automated system that can efficiently analyze resumes, understand semantic information, and accurately match candidates to job roles, improving both the speed and quality of the hiring process.

II. LITERATURE REVIEW

The reviewed studies indicate that traditional recruitment systems are inefficient and biased, while AI-based approaches significantly improve accuracy, speed, and scalability. Most modern systems rely on NLP techniques, feature extraction methods, and similarity measures to enhance candidate-job matching.

Khatri et al. [1] proposed an automated resume screening system using NLP and Machine Learning techniques. Their system extracts key information such as skills, education, and experience using preprocessing techniques like tokenization and stop-word removal. TF-IDF is applied for feature extrac-tion and similarity measures are used to match resumes with job descriptions. However, its performance is highly dependent on the quality and structure of the input data.

Desai et al. [2] developed a web-based platform to automate resume screening using TF-IDF and cosine similarity. The system significantly reduces screening time and improves efficiency, but has limitations in understanding deep semantic meaning beyond keyword similarity.

Shyamala et al. [3] proposed an intelligent recruitment system utilizing both Machine Learning and NLP techniques, applying tokenization, stop-word removal, and lemmatization along with ML algorithms to classify and rank candidates. However, the system may face challenges in handling highly complex or non-standard resume formats.

Sai et al. [4] developed a resume screening system combin-ing traditional ATS methods with NLP techniques, aiming to overcome keyword-matching limitations. The system enhances accuracy but still partially depends on keyword-based filtering. Anusha et al. [5] presented a resume screening system using ML classification and ranking algorithms, reducing manual effort and improving consistency. However, performance de-

pends on the quality and size of the training dataset.

Saatci et al. [6] applied NLP techniques to improve screen-ing efficiency and fairness, reducing human bias and improv-ing processing speed. The system requires high-quality and well-formatted data for accurate results.

Younes et al. [7] proposed an advanced recruitment system using Large Language Models (LLMs) for resume parsing, feature extraction, and semantic analysis, enabling more accu-rate matching and scalability but requiring high computational resources.

Singh et al. [8] proposed a machine learning approach classifying resumes into job categories using Naive Bayes and SVM, though focusing more on classification than detailed job matching.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

Kumar et al. [9] applied BERT for resume screening, capturing contextual and semantic meaning more effectively than traditional methods, but requiring high computational power and large datasets.

Reddy et al. [10] proposed an ontology-based approach for improved semantic job matching, capturing relationships between skills and roles, but requiring complex domain-specific ontology design.

Patel et al. [11] combined data mining with NLP, applying clustering and classification to group candidate profiles and identify suitable matches, enabling pattern discovery from large datasets.

Verma et al. [12] developed a hybrid recommendation system combining content-based filtering and collaborative filtering, improving personalization and matching accuracy but depending on historical data availability.

Gupta et al. [13] proposed an automated resume parsing system using NLP and named entity recognition to extract structured information, focusing on information extraction rather than advanced job matching.

Mehta et al. [14] introduced a deep learning-based approach using neural networks to capture semantic relationships be-tween resumes and job descriptions, providing personalized recommendations with better precision and recall.

Roy et al. [15] proposed an automated hiring system lever-aging NLP and semantic similarity using word embeddings to capture contextual meaning, improving candidate-job match-ing accuracy and ranking consistency.

Despite these contributions, challenges remain in data qual-ity, bias in training data, and limited contextual understanding, providing scope for further research and improvement.

III. SYSTEM ARCHITECTURE

The proposed AI-based ATS follows a modular architecture, where each module performs a specific function, ensuring scalability, flexibility, and ease of maintenance.

A. Overall Architecture

The system follows a pipeline architecture consisting of five key stages: input, preprocessing, feature extraction, matching and ranking, and output. The complete workflow begins at the Input Module, where recruiters upload resumes in PDF or DOCX format along with a job description. The data then flows through preprocessing, feature extraction, and similarity computation before producing ranked output results.

B. Modules of the System

1) Input Module: The Input Module is the entry point of the system, accepting a single resume (PDF/DOCX) and a job description provided by the user. It performs initial validation to check

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

file format, file size, and completeness of the job description, ensuring only valid inputs are forwarded.

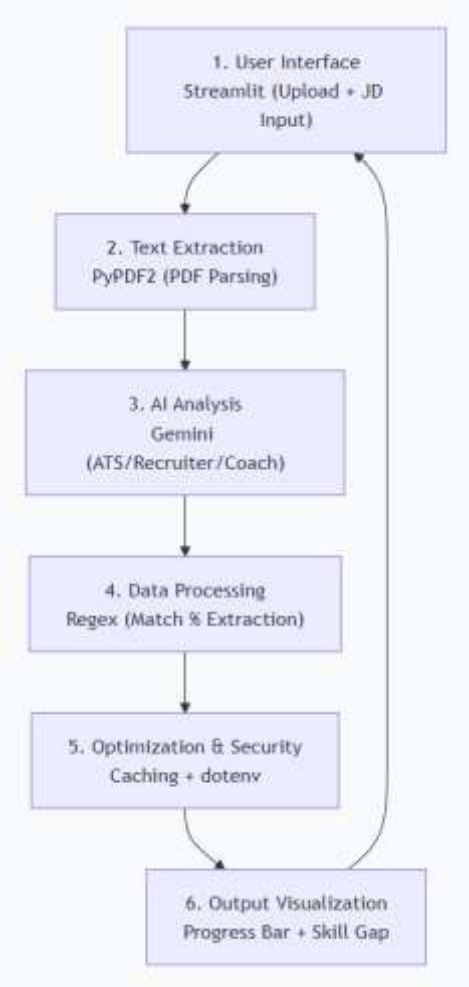

Fig. 1. Data Flow of the Proposed AI-Based ATS System

2) Resume Parsing Module: This module converts the uploaded resume into structured, machine-readable text. Using NLP-based parsing techniques, it extracts important informa-tion such as candidate name, skills, educational qualifications, work experience, and certifications, organizing them in a structured format for further analysis.

3) Text Preprocessing Module: The extracted text under-goes preprocessing to remove noise and standardize the data. The following NLP techniques are applied:

- Tokenization: Splitting text into individual words or tokens.
- Stop-word Removal: Eliminating common words such as “the,” “is,” and “and.”
- Lemmatization: Reducing words to their base form (e.g., “running” → “run”).
- Special Character Removal: Removing punctuation and irrelevant symbols.

4) Feature Extraction Module: Meaningful features are ex-tracted from both the resume and job description using TF-IDF (Term Frequency–Inverse Document Frequency) vectorization. TF-IDF is defined as:

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

$$F\text{-}IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

where TF(t, d) is the frequency of term t in document d, and IDF(t) = log N measures how rare term t is across the corpus of N documents. Both the resume and job description are converted into numerical vectors for comparison.

5) Matching Module: The Matching Module compares the processed resume with the job description using cosine similarity:

$$\text{Cosine Similarity} = \vec{A} \cdot \vec{B}$$

$$|\vec{A}||\vec{B}| \quad (2)$$

where $\vec{A}$ and $\vec{B}$ represent the TF-IDF vectors of the resume and job description respectively. The result ranges from 0 to 1, where a value closer to 1 indicates a higher degree of similarity. Cosine similarity is effective in text analysis as it focuses on vector orientation rather than magnitude, ensuring reliable comparison regardless of document length.

6) Ranking and Scoring Module: The similarity score is converted to a percentage and used to classify candidates. Predefined thresholds categorize candidates as “Fit,” “Mod-erately Fit,” or “Not Fit” for the job role. The module also highlights matched and missing skills, providing transparent and actionable insights for recruiters.

7) Output Module: The Output Module presents final re-sults through a user-friendly interface displaying the match percentage, matched skills, missing keywords, and improve-ment recommendations.

IV. METHODOLOGY

The proposed system follows a structured methodology to automate resume screening and job matching. The complete pipeline is described below.

A. Data Collection

The system accepts a single resume uploaded in PDF or DOCX format, along with a job description provided by the recruiter. Input validation ensures file format compliance, readability, and completeness of the job description before proceeding to the next stage.

B. Data Preprocessing

Raw textual data from both the resume and job description undergoes the following preprocessing steps:

1) Tokenization: The text is broken into individual tokens for granular analysis.

2) Stop-word Removal: Common words with low discrim-inative value are eliminated.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

3) Lemmatization: Words are reduced to their base form to ensure consistency (e.g., “developers,” “developing,” and “developed” all map to “develop”).

4) Normalization: Text is converted to lowercase; punctu-ation and special characters are removed.

C. Feature Extraction (TF-IDF)

After preprocessing, both the resume and job description are transformed into numerical vectors using TF-IDF vec-torization (Equation 1). This approach highlights unique and relevant terms such as specific technical skills or domain-related keywords while reducing the weight of terms that appear frequently across documents.

Word embeddings such as Word2Vec can alternatively be used to capture semantic relationships between words, plac-ing semantically similar words closer in vector space (e.g., “developer” and “programmer”).

D. Similarity Calculation (Cosine Similarity)

The cosine similarity between the resume and job descrip-tion vectors is computed using Equation 2. This metric reliably measures textual alignment regardless of document length differences, making it suitable for comparing short resumes with detailed job descriptions.

E. Ranking and Classification

The similarity score is expressed as a match percentage. Candidates are classified using predefined thresholds:

•Score ≥ 70%: “Fit” for the job role.

•40% ≤ Score < 70%: “Moderately Fit.”

•Score < 40%: “Not Fit.”

Additional outputs include matched skills and missing key-words, providing transparent and actionable insights.

F. API Integration

The system integrates the Google Generative AI (Gemini API) to enhance analysis beyond traditional TF-IDF. The API enables deeper semantic understanding, generation of resume summaries, identification of skill gaps, and improvement sug-gestions, resulting in more meaningful candidate evaluation.

G. Tools and Technologies

The system is built using the following stack:

•Programming Language: Python

•Frontend: Streamlit (interactive web interface)

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

•NLP Libraries: NLTK (tokenization, stop-word removal, lemmatization), spaCy (named entity recognition)

•ML Library: Scikit-learn (TF-IDF vectorization, cosine similarity)

•File Handling: PyPDF2, pdfplumber (PDF extraction), python-docx (DOCX extraction)

•AI API: Google Generative AI (Gemini API)

V. IMPLEMENTATION AND RESULTS

A. Implementation

The system is implemented in Python and deployed through a Streamlit-based web interface. The development workflow follows these steps:

1) Environment Setup: Installation of required libraries including NLTK, Scikit-learn, PyPDF2, python-docx, and Streamlit.

2) File Handling: Users upload resumes in PDF or DOCX format; text is extracted using appropriate parsing li-braries.

3) Preprocessing Pipeline: Tokenization, stop-word re-moval, and lemmatization are applied to both resume and job description texts.

4) Feature Extraction: TF-IDF vectorization converts pre-processed text into numerical vectors.

5) Similarity Computation: Cosine similarity computes the match score between resume and job description vectors.

6) Interface: A Streamlit interface provides file upload, text input, and result display functionality.

B. Working Demonstration

The system operates in seven steps:

1) Launch: The Streamlit interface opens in the browser with clearly labeled input options.

2) Resume Upload: The user uploads a resume (PDF/DOCX) through a file upload button.

3) Job Description Input: The recruiter enters or pastes the job description into a text box.

4) Text Extraction and Processing: The system extracts text from the resume and applies the full NLP prepro-cessing pipeline.

5) Feature Conversion: Both texts are converted to TF-IDF numerical vectors.

6) Similarity Calculation: Cosine similarity produces a match score.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

7) Result Display: The match percentage, matched skills, and missing keywords are displayed on the interface.

C. Results and Evaluation

The system was tested using multiple resumes and job descriptions across diverse job roles. Key findings are sum-marized in Table I.

TABLE I. EVALUATION OF THE PROPOSED ATS SYSTEM

Evaluation Criterion	Performance
Matching Accuracy Processing Time (per resume) Bias Reduction Consistency of Evaluation User Interface Usability Scalability (current)	Good (data-dependent) Within seconds Significant (data-driven) High (deterministic) Simple & Intuitive Single resume per query

Results confirm that candidates with relevant skills and experience receive higher match percentages, while those with fewer matching keywords receive lower scores. The system also provides matched skills and missing keyword insights, adding value beyond simple scoring. Processing is completed within seconds, significantly faster than manual review.

VI. ADVANTAGES AND LIMITATIONS

A. Advantages

1) Time Efficiency: Resumes are processed within sec-onds, compared to hours or days in manual screening.

2) Reduction of Manual Effort: Repetitive screening tasks are automated, freeing recruiters for higher-value activities.

3) Improved Accuracy: TF-IDF and cosine similarity pro-vide precise, data-driven match scores.

4) Consistency: Every resume is evaluated using the same algorithm, eliminating variation between reviewers.

5) Bias Reduction: Evaluation is based solely on skills, qualifications, and experience, not personal characteris-tics.

6) User-Friendly Interface: The Streamlit interface re-quires no technical expertise to operate.

7) Insightful Output: Matched and missing skill analysis provides actionable feedback for both recruiters and candidates.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

8) Scalability: The modular architecture supports extension to multi-resume and multi-role processing.

9) Cost-Effectiveness: Automation reduces the need for large screening teams.

10) Easy Integration: Compatible with external AI APIs, job portals, and cloud platforms for enhanced capabilities.

B. Limitations

1) Keyword Dependence: TF-IDF and cosine similarity rely on term overlap; candidates using different terminology or synonyms may receive lower scores despite being qualified.

2) Limited Contextual Understanding: The system may not fully capture the context or intent of experience descriptions; soft skills and deeper semantic relationships are not fully evaluated.

3) Single Resume Processing: The current system pro-cesses one resume at a time, limiting bulk recruitment efficiency.

4) Complex Format Handling: Non-standard resume formats with tables, columns, or images may not be accurately parsed.

5) Input Quality Dependency: System accuracy is directly linked to the quality of the input resume and job description.

6) Data Privacy Concerns: Integration with external APIs introduces data security considerations requiring proper encryption and access controls.

VII. DISCUSSION

The experimental results confirm that the proposed AI-based ATS system offers significant improvements over traditional manual recruitment. The use of TF-IDF and cosine similarity provides fast, consistent, and objective candidate evaluation. Compared with existing systems, which may achieve superior semantic understanding using BERT or LLMs [7], [9], the proposed system offers competitive accuracy with lower computational requirements, making it practical for small-to-medium organisations.

A key trade-off is observed between keyword-based matching accuracy and deeper semantic understanding. While the system effectively identifies candidates with strong keyword alignment, candidates who express equivalent skills using different terminology may be undervalued. Future integration of word embeddings or transformer-based models could address this limitation.

The system compares favourably to traditional ATS platforms on speed, consistency, and fairness. Human recruiters retain an advantage in evaluating soft skills, emotions, and contextual nuance; therefore, the system is best positioned as a decision-support tool complementing human judgment rather than re-placing it entirely.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented an AI-Based Automated Resume Screening and Job Matching System using NLP and Machine Learning. The system addresses major challenges in traditional hiring, including time consumption, manual effort, inconsistency, and human bias. By integrating TF-IDF vectorisation and cosine similarity within a modular NLP pipeline, the system efficiently analyses resumes and generates match scores with skill-level insights.

The system achieves fast processing (within seconds per re-sume), consistent and objective evaluation, and a user-friendly Streamlit interface accessible to non-technical recruiters. Integration with the Google Gemini API further enhances semantic analysis capabilities. These results demonstrate the practical viability of AI-driven automation for improving recruitment efficiency, accuracy, and fairness.

B. Future Work

Several directions can further enhance the system:

•Deep Learning Integration (BERT, LLMs): Replacing TF-IDF with transformer-based models to capture deeper semantic meaning and handle synonym-rich resumes more accurately.

•Multilingual Support: Integrating language detection and translation APIs to process resumes in multiple languages, expanding applicability across regions.

•Real-Time Job Portal Integration: Connecting to job portals for automated job description retrieval and real-time candidate matching.

•AI-Based Interview System: Extending the pipeline with AI-driven chatbot or video interview systems for initial candidate evaluation beyond resume screening.

•Bulk Resume Processing: Scaling the system to simultaneously process and rank multiple candidates for large-volume recruitment.

•Advanced Bias Detection: Incorporating fairness-aware ML models to further reduce and audit bias in the evaluation process.

REFERENCES

[1] Khatri et al., “Automated Resume Screening and Job Matching System Using NLP,” 2025.

[2] Desai et al., “A Web-Based AI Recruitment System Using TF-IDF and Cosine Similarity,” 2025.

[3] Shyamala et al., “Smart Recruitment System Using Machine Learning and NLP,” 2026.

[4] Sai et al., “Resume Analysis System Using ATS Algorithms,” 2026.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 872-883

[5] Anusha et al., “Machine Learning-Based Resume Screening System,” 2025.

[6] Saatci et al., “Resume Screening Using NLP Techniques,” 2024.

[7] Younes et al., “AI-Based Applicant Tracking System Using Large Language Models,” 2025.

[8 Singh et al., “Resume Classification Using Machine Learning,” 2023.

[9] Kumar et al., “Deep Learning-Based Resume Screening Using BERT,” 2024.

[10] Reddy et al., “Semantic Job Matching System Using Ontology-Based Approach,” 2022.

[11] Patel et al., “Intelligent Recruitment System Using Data Mining and NLP,” 2023.

[12] Verma et al., “Hybrid Recommendation System for Job Matching,” 2023.

[13] Gupta et al., “Automated Resume Parsing Using NLP,” 2021.

[14] Mehta et al., “Deep Learning-Based Job Recommendation and Resume Matching System,” 2024.

[15] Roy et al., “Automated Hiring System Using Semantic Similarity and NLP,” 2024.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” arXiv preprint arXiv:1301.3781, 2013.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL, 2019.

[18] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” Information Processing & Management, vol. 24, no. 5, 1988.

[19] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, 2011.

[20] A. Vaswani et al., “Attention Is All You Need,” in Proc. NeurIPS, 2017.