ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN: 2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

Resume Rank AI: AI-Powered Resume Screening and Candidate Ranking

[1]Adama Yogitha, [2]Mrs. K Sheetal

[1]M.Tech Scholar, Dept. of CSE (AI&ML), Malla Reddy Technical Campus, Malla Reddy Vishwavidyapeeth (Deemed to be University), Maisammaguda, Hyderabad, Telangana 500100, India, adamayogitha@gmail.com

[2]Assistant Professor, Dept. of CSE, Malla Reddy Technical Campus, Malla Reddy Vishwavidyapeeth (Deemed to be University), Maisammaguda, Hyderabad, Telangana 500100, India. sheetalkulkarni925@gmail.com

Abstract

Traditional resume screening is subjective, time-consuming, and vulnerable to human bias, which causes inconsistency in the recruitment process. To get over these limitations, this project proposes a system that uses AI methods including automated ranking algorithms, machine learning, and natural language processing (NLP) to rate applicants and screen resumes. A candidate's suitability for a job is determined by the system's analysis of their resume, which takes into account their education, certifications, work experience, and abilities. The data is then compared to a database that contains established job requirements. Using complex rating and classification algorithms, the model improves decision-making and reduces the recruiter's workload. The proposed method employs supervised learning strategies, feature engineering, and data preprocessing to build a powerful ranking framework. It also includes candidate rating methods to guarantee an honest evaluation. Our technique, in contrast to conventional keyword-matching approaches, places more emphasis on contextual analysis of candidate profiles in order to better align credentials with job descriptions. Results from the training phase demonstrate the model's effectiveness in real-world recruitment procedures, with outstanding accuracy in resume categorization and candidate rating. The system's ultimate purpose is to increase data-driven hiring decisions, decrease bias in the hiring process, and streamline the recruitment process; it is an important tool for modern HRM.

Introduction

Because it affects growth, productivity, and competitiveness, hiring the correct people has long been a crucial aspect of human resource management. There is now an overwhelming number of applications for every available job due to the meteoric rise of online job portals and career platforms. Recruiters may get hundreds, if not thousands, of resumes for a single post, making manual screening very inefficient, unequal, and error-prone. While conventional methods of hiring were effective in bygone decades, they simply cannot handle the sheer volume and complexity of today's application evaluations. Even with all the information they have, human recruiters still have constraints in their brain capacity and time. As a result, they could overlook qualified applicants or even introduce bias.One interesting development in response to these issues is the incorporation of AI into the hiring process. Computer programs powered by AI can swiftly sort through vast amounts of resumes, find the ones that are relevant, and rate them based on how well they match the position. In contrast to human screening, which relies heavily on subjective judgment, screening powered by artificial intelligence (AI) ensures consistency, equality, and data-driven decision-making. Employing ML and NLP, the system can go further into resume content than only searching for keywords. It can then determine how relevant a person's qualifications, experiences, and credentials are in a certain setting.In the hiring process, AI may be a powerful tool because of its ability to increase efficiency while decreasing bias. Multiple organizations have concluded that human bias is the root cause of most forms of discrimination, including those based on gender, color, and level of education.

If AI systems are well-designed, they may narrow their focus to the specific skills and credentials required for a job, perhaps eliminating or significantly reducing such biases. Also, by analyzing resumes, developing shortlists of possibilities, and filtering applications for unnecessary material, AI might significantly save human work. Human recruiters may then focus on more nuanced factors, such cultural fit and interpersonal skills.Building and releasing an AI-

ISSN: 2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

powered system for reviewing applications and assessing applicants is the main focus of this project. To process resumes, one must first convert data structures, automate feature extraction, and then map features against job requirements. The system evaluates candidates based on how well they fit the job's criteria using machine learning categorization models. Finding the top candidates quickly could end up saving recruiters a lot of time and money.Furthermore, this initiative places a premium on transparency in the evaluation process. This approach does not function as a mysterious "black box," but rather produces reliable and understandable results for scoring and ranking. The development architecture combines supervised learning approaches with ranking algorithms to ensure accurate candidate categorization and prioritization. Additionally, the system is able to manage irregular wording, extraneous content, and various resume formats via the use of strong data preparation algorithms.Incorporating such a system has the potential to radically alter HR practices. With more and more recruitment happening online, clever solutions that can adapt to shifting job requirements and candidate pools are essential for organizations. This project offers a practical solution for industries seeking scalable, unbiased, and efficient recruitment processes. It also contributes to academic research on AI-driven recruitment.

Literature Survey

We provide a workable method for extracting latent talents from job postings and unstructured resumes by using document embedding techniques. The authors detail a process for transforming job postings and resumes into dense vector representations, which will allow for more convenient semantic comparisons than simple keyword matching. They show that embeddings improve matching accuracy in job recommendation tasks by discovering latent relationships between talents, projects, and jobs. The approach proceeds to supervised fine-tuning for downstream suggestions once the embeddings are constructed unsupervised. The study states that compared to TF-IDF baselines, experimental findings show better retention of important abilities and greater match rates. The article goes on to describe the benefits of skill semantics, elaborating on how they may be used to find hidden or unusual abilities that aren't easy to find using keyword searches. Included in the scope of this article are issues related to preprocessing, tokenization, and handling noisy resume forms. Working with a database of job ads and resumes, the authors evaluate the method and look at instances when domain-specific words led the model astray. They suggest hybrid approaches that combine document embeddings with explicit entity extraction after finding that embeddings greatly improve job-resume matching.

In order to scan resumes and rank them against job descriptions, this study seeks to assess an automated system that utilizes transformer-based contextual embeddings (BERT). The authors enhance a BERT model by obtaining appropriateness ratings from a tagged resume-job dataset. After that, they show that contextual encodings are better than the traditional bag-of-words and TF-IDF tactics. The whole pipeline is described in the study, beginning with resume cleaning and ending with supervised grading, aggregating methods, and sentence encoding. Research shows that people are better able to remember and accurately choose the top k candidates, as well as analyze references to subtle skills and dynamic language. As an additional means of elucidating the factors considered in the ranking, the authors address model explainability methodologies. They propose a method they term production-friendly lightweight distillation to deal with real-world problems including computational expense and the need for labelled data for adjustment. The outcomes of input length reduction and pooling procedures have been investigated empirically via studies referred to as ablation research. The study claims that when the event's context is crucial, BERT-style models perform well.

A lot of prior work has gone into developing resume parsers that can make sense of semi-structured resumes and pull out the usual structured data like name, contact, education, skills, and experience. The authors provide a technique for normalizing data for application monitoring databases using pattern matching, statistical NER, and rule-based extraction. These algorithms include techniques for evaluating the parsing accuracy of various formats (PDF, DOCX, and plain text), normalizing dates, and distinguishing sections (such as "Education" and "Experience"). Findings show that various resume formats may be successfully retrieved using lightweight ML to manage noisy portions and deterministic criteria. Also covered in depth are issues like embedded tables, noisy OCR output, and mis-parsed multi-column formats. Data is sent via a parser as a pre-processing step before it is provided to ranking modules and classifiers. The research offers assessment measures

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

for field extraction, including as recall and accuracy, and proposes iterative human-in-the-loop correction for bootstrap parsers for new domains. In light of these results, resume-driven systems may benefit from hybrid rule+ML parsing.

Methodology

The proposed system's methodology consists of several sequential processes, the first of which is data collection. We take a variety of resume formats and standardize them for uniform processing. Next, we clean the text by removing stop words, special characters, and redundant material while maintaining relevant keywords and characteristics. This is known as data preparation. To create structured feature representations from unstructured text, the system employs a variety of natural language processing (NLP) techniques, such as tokenization, lemmatization, and vectorization. Once features have been retrieved, feature engineering may identify significant attributes such as skills, education, work experience, and certifications. When these characteristics are matched with predefined job requirements, it produces candidate-job match scores. In order to classify data, machine learning algorithms scour labelled datasets for patterns indicating which candidates are most suitable. Methods like neural networks, decision trees, or logistic regression might be used here. After that, an algorithm is used to rank the candidates based on how relevant they are projected to be. Model evaluation using performance measures such as F1-score, accuracy, precision, and recall is also a part of the method for evaluating effectiveness. A straightforward scoring interface allows recruiters to assess candidate rankings and suitability explanations. That way, we know the approach is practical for real-life hiring.

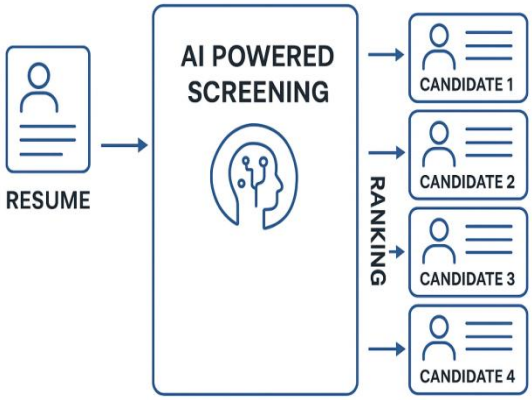


Fig: Proposed system Block diagram

To rate applicants and filter resumes using AI, we provide an automated, scalable, and smart system. The flaws and prejudices in conventional hiring methods are what this methodology is trying to address. A preprocessing pipeline is the backbone of the system; it cleans and standardizes the incoming data using digital resumes. Utilizing advanced natural language processing techniques to extract structured data from unstructured resume material, the system identifies critical components such as a candidate's skills, job history, education, and certifications. The applicant's suitability for the position is determined by comparing their attributes to those listed in the job description. The method employs supervised machine learning algorithms to categorize candidates into three groups: "very suitable," "moderately suitable," and "less suitable." The method goes beyond simple candidate categorization; it also employs a grading system to generate a priority list, with the more relevant ones given precedence. By combining rating and classification, recruiters may significantly cut down on the time spent on initial screening, leading to much more efficient recruitment. The proposed method can analyze resumes contextually, which is a significant improvement above relying just on keyword matching. When evaluating a programmer's skills, the system considers more than just their programming language knowledge; it also looks at their project experience and their familiarity with domain-specific applications. This contextual analysis ensures that applicants' skills are accurately evaluated.

The proposed methodology encourages transparency and equity by providing recruiters with reasons for

ISSN: 2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

application rankings using interpretable score systems. The system's capacity to graphically depict the worth of attributes and their relationship to job requirements eliminates the "black box" issue that often occurs with AI applications. On top of that, the system is designed to process many resumes simultaneously from different job ads. Thanks to its compatibility with existing recruitment platforms and applicant tracking systems (ATS), the system may be readily deployed by organizations. Machine learning (ML), ranking algorithms, and natural language processing (NLP) allow us to assess candidates with high precision and little human error. The proposed approach streamlines the recruiting process while promoting equitable hiring practices, which is a huge plus for modern human resource management.

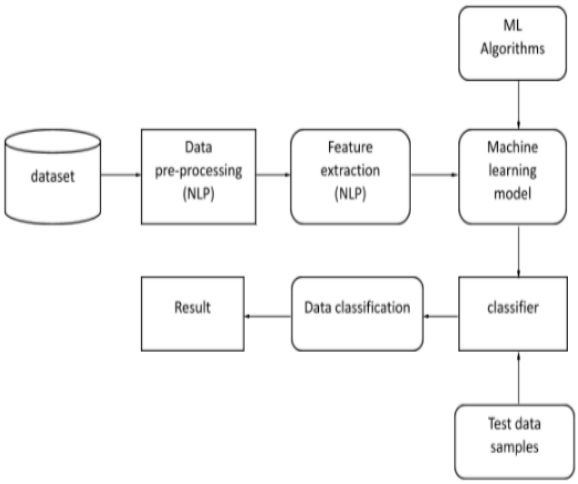


Fig: System architecture

A model of a text classification system using machine learning and natural language processing (NLP) is presented as the system architecture. Collecting relevant data sets, or datasets, for use in training and testing the model is the first phase. Sentences, papers, reviews, social media posts, and similar types of unstructured material are often found in this collection. The first step of the design is to preprocess the data using techniques from natural language processing. Data pre-processing is crucial because unprocessed data often contains noise, irrelevant information, stopwords, punctuation, and errors, all of which may reduce the accuracy of the model. As part of the pre-processing, the text may be tokenized, removed common words like "the" and "is," reduced to their root form through stemming or lemmatization, lowercased to ensure uniformity, and handled missing or inconsistent values among other things.

In order to get the input text ready for processing, it is cleaned up and arranged by pre-processing. The next step after pre-processing is feature extraction by use of natural language processing techniques. The feature extraction process involves transforming the cleaned textual input into numerical representations that may be understood by machine learning models. Machine learning algorithms can only interpret words once they are converted into meaningful numerical vectors. Some examples of advanced word embeddings include Word2Vec, GloVe, and BERT. Two popular feature extraction approaches are TF-IDF and BoW. By capturing the syntactic and semantic links of word associations, these approaches enhance the quality of the input features.

Text categorization relies heavily on feature extraction, which ensures that the algorithm will get valuable information from text. Once feature extraction is complete, the processed features are sent on to the machine learning model. Training a model using targeted machine learning techniques is the meat and potatoes of this architectural step. Which ML approaches are used depends on factors including the quantity of the dataset, the necessity for classification, and the kind of problem. These could include CNNs, RNNs, and other deep learning models, together with algorithms such as Naïve Bayes, Logistic Regression, Support Vector Machines, Decision Trees, and Random Forests. The selected algorithm learns to interpret the collected features by looking for trends and correlations during training.

This trained machine learning model is the decision-making engine that looks forward to seeing data samples in order to make predictions. Machine learning methods may be used to the characteristics that have been retrieved, enabling the model to generalize from the training dataset. It is at this point that the model is constructed. This stage involves optimizing the model via training iterations, hyperparameter tweaking, and cross-validation in order to maximize accuracy and prevent overfitting. Following its initialization, a classifier is included into the machine learning model. A classifier is a module for generating decisions that labels new data samples using training data. It uses the trained model to classify input samples into predefined buckets according to characteristics like spam status, sentiment (positive or negative), or domain-specific labels. All the while, the design incorporates test data samples to assess the trained classifier.

The only way to objectively evaluate the model's performance is to separate the test data from the training dataset. By giving the classifier test data

ISSN: 2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

samples, we may assess its prediction capability and generalizability. Memory, correctness, precision, F1-score, and confusion matrix are common measures used to evaluate classification performance. To measure the classifier's efficacy, we compare its predicted labels for the test samples to the real labels. Following classification, the procedure proceeds based on the results of data classification. This is the stage when we decipher the classifier's output. The correct mapping, storage, and analysis of the predicted categories for further insights rely heavily on the data categorization step. The unprocessed output from the classifiers is transformed into meaningful classifications that align with the challenge objectives by this step.
For example, in a sentiment analysis task, the data classification stage may label reviews as positive, neutral, or negative depending on the classifier's output. The final product of the pipeline is represented by the result, which is then generated by the system. The report, data set, or anticipated label that comes out of it all depends on the application. Visual representations of the findings, such as charts or graphs, may help users grasp the meaning of the data. This architecture is successful because it uses test data fairly, the machine learning algorithm is resilient, features are plentiful, and preprocessing is of high quality.

This design follows a logical progression from the first data collection phases all the way to the generation of valuable outputs by means of automation and natural language processing. Preprocessing, feature extraction, model training, input categorization, and performance testing are all emphasized as crucial steps in the process of cleaning up raw text and converting language into numbers. Developed for use in a wide range of natural language processing (NLP) classification tasks, this method is adaptable, precise, and extensible. The system is also adept at handling large volumes of unstructured data because to the combination of machine learning and natural language processing (NLP). Practical uses of these ideas include building chatbots, implementing information retrieval systems, translating languages, analyzing sentiment, and detecting spam. By following this pipeline, organizations may create robust AI solutions that can transform unstructured text into actionable insights, potentially improving their decision-making processes across industries.

Implementation

DATA COLLECTION

The data used in this work is derived from JM1 software. Picking out the data subset that will serve as your basis for analysis is the focus of this stage. To begin solving ML issues, you need data—ideally, a large amount of data in the form of instances or observations for which the desired outcome is known. Labeled data is information for which the desired outcome is known in advance.

DATA PRE-PROCESSING

Sort the data you've chosen by cleaning, formatting, and sampling. These are three typical procedures for pre-processing Data format: It's possible that the format you've chosen isn't ideal for working with the data you've chosen. The data may be stored in a proprietary file format that you would want converted to a relational database or text file, or it could be in a relational database that you would like in a flat file format. The process of cleaning data involves erasing or correcting any data that is missing. You may not have all the information you need to fix the issue if certain data instances are missing or incomplete. Eliminating these cases could be necessary. Furthermore, some characteristics could include sensitive information, which necessitates either anonymizing or removing them from the data completely.Sampling: You can find that there is a lot more accessible sampled data than you really need. Increases in both computational and memory demands, as well as algorithm execution durations, are possible outcomes of data explosion. Before diving into the whole dataset, it could be helpful to pick a smaller, more representative sample of the chosen data. This will allow you to explore and prototype ideas more quickly.

FEATURE EXTRATION

Feature extraction, a technique for decreasing the amount of characteristics, should be performed next. While feature selection sorts features according to their predictive relevance, feature extraction alters the traits themselves. When traits or attributes change, they are really just linear combinations of their original values. To train our models, we ultimately use the Classifier technique. We utilize the classify module from the Natural Language Toolkit Python package. We used the labeled dataset that had been gathered. We will use the remaining tagged data to test the models. A few of machine learning algorithms were

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

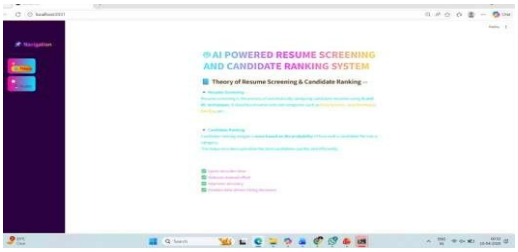
ISSN: 2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

used to categorize previously processed data. Classifiers from the random forest were chosen. There is a lot of use for these methods in text classification difficulties.

EVALUATION MODEL

An essential aspect of developing a model is evaluating it. Finding the optimal model to describe our data and gauging the model's future performance are both aided by this. Due to the high likelihood of producing too optimistic and overfit models, it is unacceptable in data science to evaluate model performance using the same data used for training. We can evaluate the performance of each categorization model by averaging its results. You may expect to see the outcome in its graphic form. Visualization of data using a classification system using graphs. The proportion of right predictions made using the test data is called accuracy. Simply divide the total number of forecasts by the number of right predictions, and you get the answer.



Results

Fig: Theory Section of AI-Powered Resume Screening System

Fig: User Interface of AI-Powered Resume Screening and Candidate Ranking System

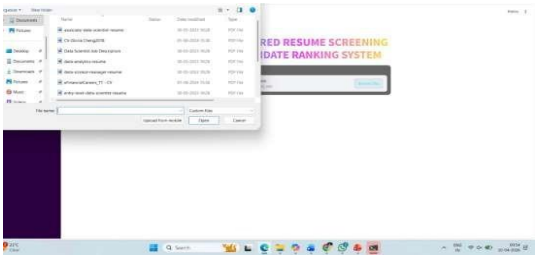


Fig: Resume Upload Interface of AI-Powered Screening System

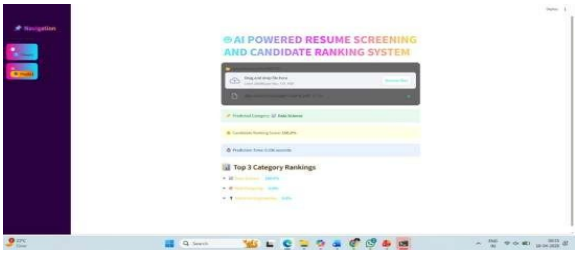


Fig: Resume Analysis and Candidate Ranking Result

Conclusion

This experiment's results demonstrate the potential game-changing impact of AI-driven application rating on the recruitment process. Through the integration of automated ranking methodologies, machine learning algorithms, and Natural Language Processing (NLP), the system provides a robust framework for reviewing resumes and rating applicants. By eschewing traditional keyword-based methods in favor of contextual information about skills, experience, and education, this approach produces more precise suitability evaluations. The results demonstrate that the AI approach ensures evaluation fairness, reduces the workload for recruiters, and increases classification accuracy. Because it gives proof for data-driven decisions and aids in reducing bias in the hiring process, it is an essential tool for modern HRM. There are several possible applications for the method, including government recruitment, university placements, and freelancing platforms; it represents a significant advance in the search for smarter, more efficient, and more egalitarian recruiting processes. When it comes to hiring new employees, artificial intelligence (AI) will revolutionize the industry by helping businesses save time, save costs, and increase efficiency without compromising on quality.

References

[1] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT networks," in *Proc. EMNLP*, 2021.

[2] Y. Zhang *et al*., "Skill extraction from job postings using BERT," in *Proc. Int. Conf. NLP Applications*, 2021.

[3] S. Kumar *et al*., "Job recommendation system using machine learning," *Journal of Artificial Intelligence Research*, 2021.

[4] X. Li *et al*., "Neural job matching with deep learning," *IEEE Trans. Knowledge and Data Engineering*, 2021.

[5] R. Sharma *et al*., "Resume screening using NLP and machine learning," *International Journal of Computer Science*, 2021.

[6] A. Gupta *et al*., "Transformer-based resume classification," in *Proc. IEEE Conf. Data Mining*, 2022.

[7] L. Chen *et al*., "Job recommendation using hybrid deep learning models," in *Proc. ACM Conf. Recommender Systems*, 2022.

[8] P. Singh *et al*., "Named entity recognition for resume parsing using BERT," in *Proc. IEEE Conf. NLP Systems*, 2022.

[9] X. Wang *et al*., "Resume ranking using learning-to-rank models," in *Proc. SIGIR*, 2022.

[10] H. Zhao *et al*., "Skill matching using knowledge graphs," *IEEE Access*, vol. 10, 2022.

[11] Y. Liu *et al*., "Deep learning-based talent recommendation system," *ACM Trans. Information Systems*, 2023.

[12] N. Patel *et al*., "Context-aware resume screening using NLP," *Springer Journal of Data Science*, 2023.

[13] S. Verma *et al*., "Automated resume screening using BERT and XGBoost," in *Proc. IEEE Int. Conf. AI*, 2023.

[14] J. Kim *et al*., "Explainable AI for resume screening," *Journal of Machine Learning Research*, 2023.

[15] R. Roy *et al*., "Multi-modal resume screening using text and metadata," in *Proc. ACL*, 2023.

[16] OpenAI Researchers *et al*., "Skill extraction using large language models," *arXiv preprint*, 2024.

[17] P. Desai *et al*., "AI-based recruitment system using transformers," in *Proc. IEEE Conf. Intelligent Systems*, 2024.

[18] M. Ahmed *et al*., "Fairness-aware resume screening using machine learning," *Journal of AI Ethics*, 2024.

[19] K. Rao *et al*., "Resume matching using semantic embeddings and cosine similarity," *International Journal of Data Science*, 2024.

[20] T. Brown *et al*., "LLM-based job recommendation systems," in *Proc. NeurIPS Workshops*, 2024.