INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 11, November 2026 | ISSN: 2456-4184 | IJNRD.ORG

A SURVEY ON AI-BASED RESUME PARSING AND AUTOMATED CANDIDATE SCREENING

Minal Dharne[1], Priyanka Shahane[2]
[1]Student, Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune, Maharashtra, India
[2]Assistant Professor, Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune, Maharashtra, India

Abstract— Recruitment organizations receive a large number of resumes for a single job opening, making manual screening time-consuming and difficult to perform consistently. Traditional keyword-based filtering may overlook suitable candidates when equivalent skills are expressed using different terms, and it may give excessive importance to repeated or irrelevant keywords. This survey reviews artificial intelligence, natural language processing (NLP), and machine learning techniques used for resume parsing, skill extraction, job-role classification, and candidate ranking. The reviewed approaches include text preprocessing, named entity recognition, TF-IDF, cosine similarity, classification algorithms, semantic representations, and nearest-neigh bour ranking. The proposed Resume Parser AI workflow uses Python and NLTK for preprocessing, groups related skills and terms through clustering, and evaluates the relevance of a resume to a specified job description. A normalized weightage score from 0 to 10 is proposed to support recruiter review. This paper compares existing approaches, discusses evaluation measures, and identifies challenges such as resume-format variation, keyword manipulation, dataset limitations, fairness, and explainability. The score range is a design objective of the proposed system; no experimental performance result is claimed.

Keywords— Resume parsing, candidate screening, artificial intelligence, natural language processing, NLTK, skill extraction, clustering, TF-IDF, cosine similarity, resume ranking.

I. INTRODUCTION

Recruitment is an important organizational activity in which applicants are evaluated according to their education, technical skills, professional experience, projects, certifications, and suitability for a particular job role. When hundreds of resumes are submitted for one vacancy, manual examination requires considerable time and may lead to inconsistent shortlisting. An automated resume parser can convert unstructured resume documents into structured candidate information and help recruiters identify profiles that are relevant to a job description.

A resume generally contains both structured and unstructured information. Names, email addresses, education details, skills, employment history, and project descriptions may appear in different sections and formats. A single skill can also be expressed in several ways; for example, a candidate may describe experience in 'machine learning', 'ML', or a particular machine-learning library. Simple keyword matching may fail to recognize such relationships. A useful screening system should therefore consider context and relevance, rather than counting keyword occurrences alone.

Resume Parser AI aims to automate the initial screening process. The system accepts resumes and a target job position, extracts relevant text, preprocesses it, identifies skills and experience, and compares the extracted information with the job requirements. Python provides the implementation environment, while NLTK can support tokenization, stop-word removal, stemming, and other text-processing operations. Clustering can be used to group related terms or skill profiles.

The central objective is to assign each resume a normalized relevance score between 0 and 10, where 0 represents a very weak match and 10 represents a very strong match. The score is intended as a decision-support measure, not as an automatic final hiring decision. Recruiters should be able to inspect the matched skills, missing requirements, and evidence behind each score.

This survey reviews research on resume information extraction and automated job matching. It compares the methods reported in prior work and discusses how their strengths can inform a practical resume parser. The main focus is the combination of preprocessing, skill grouping, relevance calculation, and explainable ranking.

II. LITERATURE SURVEY

Automated resume screening has been studied using information retrieval, NLP, machine learning, and recommendation techniques. The reviewed studies differ in their input data, feature representation, learning algorithm, and ranking strategy. Some systems concentrate on extracting structured information from a resume, while others compare the candidate profile with a job description.

K. et al. presented a machine-learning-based resume ranking system that applies text preprocessing and TF-IDF representation to resumes and job descriptions. Cosine distance is used to estimate textual similarity, and a K-nearest-neigh bour method is used to identify and rank resumes related to the supplied job description. The study reports parsing and ranking results on its own dataset, demonstrating the use of similarity-based ranking for initial candidate screening [1].

Bardu cci et al. proposed an end-to-end framework for information extraction from Italian resumes. Their approach segments resume text into semantically consistent parts and applies named entity recognition to obtain candidate information such as personal details, skills, and work experience. The study highlights that resume heterogeneity and the absence of common document standards make extraction challenging [2].

The study titled 'Automated Resume Parsing: A Natural Language Processing Approach' combines named entity recognition with keyword and pattern matching based on regular expressions. The approach extracts entities such as names and contact information and identifies details including job titles, organizations, education, and experience. It illustrates how statistical NLP and rule-based extraction can complement each other [3].

Research on job-vacancy ranking with sentence embeddings, keywords, and named entities considers matching a candidate's curriculum vitae with employment requirements. Sentence-level representations can capture relationships that are not always visible through exact keyword comparison, while named entities and keywords provide interpretable signals for matching [4].

The IEEE conference paper 'Resume Ranking with TF-IDF, Cosine Similarity and Named Entity Recognition' discusses resume-ranking methods based on textual relevance and information extraction. TF-IDF and cosine similarity provide a baseline for comparing resumes with job descriptions, while named entity recognition can improve the identification of skills and experience. The work also draws attention to bias, transparency, and privacy concerns [5].

The study 'Optimizing Job Matching Through Automated Resume Screening with NLP and Machine Learning' describes preprocessing operations such as tokenization, stemming, lemmatization, TF-IDF, and string matching. It compares machine-learning models including SVM, logistic regression, random forest, and K-nearest neigh bours for job-role classification and reports SVM as the best-performing model in its evaluation [6].

Rathi et al. describe an NLP-powered resume-matching application that extracts text from PDF files, preprocesses the content, and compares candidate skills with a job description. Their system uses TF-IDF and cosine similarity to calculate semantic relevance and applies a trained model to estimate job fit [7].

Recent work has also explored large language models for resume parsing and job-description matching. An IEEE conference study published in 2026 evaluates GPT-based models, fine-tuning, retrieval-augmented generation, and prompt engineering using a collection of real resumes. It reports that human input alongside AI evaluation can support recruitment productivity and decision quality [8].

Chandak et al. [9] developed a resume parser integrated with a job-recommendation module. Their parser uses PDF Miner to extract text from PDF resumes, regular expressions and spaCy Matcher to identify entities and skills, and a Python-based processing pipeline. Extracted skills are passed to a recommendation component that applies TF-IDF to job titles and cosine similarity to rank related job opportunities. The work demonstrates an end-to-end connection between resume information extraction and job recommendation, although the paper does not report a single quantitative resume-parsing accuracy value for the complete system.

Jayakumar et al. [10] presented an AI-based recruitment framework that includes resume parsing, candidate screening, recommendation, and human-resource analytics. The resume parser extracts candidate details such as name, contact information, skills, and college name and stores the information for subsequent search and matching. The paper also reports an employee-attrition prediction experiment; its accuracy, precision, recall, and

F1-score refer to attrition prediction rather than resume-parsing accuracy. The candidate-matching interface reports match percentages above 80% for the top three candidates, providing a practical example of job-specific candidate ranking.

Bocharova and Malakhov [11] proposed Cap Style BERT, a BERT-based resume parsing model that incorporates capitalization and document-style features. The method represents uppercase/lowercase patterns, relative font size, and font styles such as bold and italic/underlined as additional features. The evaluation uses 7,219 resumes for the parsing task and approximately two million resumes for domain pre-training. Cap Style BERT improves the resume-segmentation F1 performance by approximately 3–6% over the domain-adapted BERT baseline, with a reported 5.4% gain in the low-resource setting using 5% of the training data.

Gaur et al. [12] introduced a semi-supervised deep-learning NER model for extracting educational institution names and degree names from resume education sections. The method begins with a manually annotated seed set, applies PDF-to-JSON conversion and NLTK-based POS tagging, predicts entities in un labelled resumes, corrects the predictions, and adds the corrected examples to the training set for retraining. With 550 seed resumes, the proposed model achieved 92.06% accuracy, 71.13% precision, 75.56% recall, and a 73.28% F1-score for the combined education-entity task.

Wos I a k et. al. [13] proposed a hybrid information-extraction framework for Polish resumes used in IT recruitment. The system combines section splitting, dictionary matching, fuzzy matching, and several NER tools, including Liner2, NERF, and Babel f y, with a CRF module. The evaluation used 1,326 structured resume files and a separate 17,000-CV collection for CRF experiments. The hybrid parser detected educational institutions at 30.92%, compared with 19.23% for the best individual NER tool, and company names at 19.68%, compared with 7.50% for the best individual tool. The study also identifies outlier handling and low-resource-language NER as continuing challenges.

War u s a with ana et al. [14] developed a layout-aware resume parser using NLP and rule-based techniques to extract complete section-wise content rather than only isolated entities. The system uses PDF layout and word bounding boxes to identify paragraph boundaries, then applies a Count Vectorizer–TF-IDF–SVM pipeline with SGD Classifier to classify resume sections, supported by custom NER and regular expressions. In the reported evaluation, the SVM model achieved an overall accuracy of 0.888, compared with 0.840 for Multinomial Naive Bayes. Section-level F1-scores and cosine similarity values indicate that performance varies by resume section, with work experience being a particularly difficult category.

Arokiaraj et al. [15] introduced the Applicant Resume Tracking System (ARTS), a generative-AI agent built with Google Gemini Pro Vision and a Stream lit interface. It accepts single or multiple PDF resumes and job descriptions, analyzes the documents, highlights candidate strengths and weaknesses, calculates a percentage match, and identifies missing skills or keywords. The paper emphasizes support for both recruiters and job seekers. It does not provide a controlled benchmark with a clearly reported parsing precision, recall, or F1-score, so its contribution is best viewed as an applied LLM-based screening and feedback workflow.

Tejaswini et al. [16] designed a machine-learning-based resume ranking system using content-based recommendation. Resume and job-description text are represented using TF-IDF, cosine similarity measures their relevance, and K-nearest neigh bours ranks candidate CVs. The paper reports an average text-parsing accuracy of 85% and a ranking accuracy of 92%. These results provide a useful quantitative baseline for similarity-based resume screening, while the authors also note the difficulty caused by non-standard resume formats.

The literature indicates that no single technique addresses every stage of the recruitment pipeline. Text extraction and NER help structure the resume; TF-IDF and cosine similarity provide efficient lexical matching; clustering can organize related skills; and semantic models can improve contextual matching. A practical system can combine these components according to its dataset size, computational resources, and explainability requirements.

III. PERFORMANCE PARAMETERS

The performance of a resume parser should be evaluated at more than one stage. Parsing quality measures whether the system extracts the correct information from the original document. Matching quality measures whether relevant skills and experience are identified for the selected job. Ranking quality measures whether the

most suitable candidates appear near the top of the generated list. The following parameters can be used when labelled evaluation data and recruiter-reviewed outcomes are available.

Precision measures the proportion of resumes predicted as suitable that are actually suitable. It is calculated as TP/(TP+FP), where TP is the number of correctly identified suitable resumes and FP is the number of unsuitable resumes incorrectly selected. High precision reduces the number of irrelevant profiles sent to recruiters.

Recall measures the proportion of all genuinely suitable resumes that the system successfully identifies. It is calculated as TP/(TP+FN), where FN represents suitable resumes that the system misses. Recall is important because a screening system should not reject qualified candidates simply because they use different wording for their skills.

The F1-score is the harmonic mean of precision and recall and is calculated as 2 × (Precision × Recall) / (Precision + Recall). It is useful when both false selections and missedcandidates matter. Accuracy is the ratio of correctly classified examples to the total number of evaluated examples: (TP+TN)/(TP+TN+FP+FN).

For ranking systems, Precision K and Recall K can measure the quality of the first K recommended resumes. Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) can assess the ordering of relevant candidates. Parsing evaluation may additionally use entity-level precision, recall, and F1-score for fields such as skills, education, and experience.

The proposed system also uses a normalized relevance score from 0 to 10. A possible scoring design combines required-skill coverage, experience relevance, education or certification relevance, and project relevance. For example, each component can be normalized to a common range and combined using weights selected with recruiter input. The weights must be documented and validated; the score should not be interpreted as a measured accuracy or probability of hiring success.

A further evaluation dimension is robustness. The parser should be tested on resumes with different layouts, file formats, abbreviations, spelling variations, and section headings. Fairness checks should examine whether irrelevant personal attributes influence ranking. Processing time and memory usage are also useful operational measures when a large number of resumes must be screened.

IV. COMPARATIVE ANALYSIS

Table I compares the previously reviewed approaches with the additional research papers supplied for this revision. The comparison records each study's principal method, supporting techniques, input data, and reported evaluation. Numerical results are reproduced only where they are explicitly reported in the corresponding paper; metrics from different tasks, such as employee-attrition prediction, are identified separately and are not treated as resume-parsing accuracy.

TABLE I. COMPARATIVE ANALYSIS OF RESUME PARSING AND CANDIDATE SCREENING TECHNIQUES

Sr. No.	Study / Approach	Best / Main Technique	Other Techniques	Dataset / Input	Evaluation Focus
1	Machine-learning-based resume ranking [1]	Cosine similarity + KNN	Text cleaning, TF-IDF	Resume and job-description collection	Parsing and ranking accuracy
2	End-to-end information extraction [2]	Named Entity Recognition	Text segmentation, linguistic patterns	Italian resumes	Entity extraction quality
3	Automated resume parsing [3]	NER + Regex	Keyword and pattern matching	Resumes in varied formats	Extraction accuracy and speed
4	Job-vacancy ranking [4]	Sentence embeddings	Keywords, named entities	CVs and job vacancies	Relevance and ranking

5	Resume ranking with TF-IDF [5]	TF-IDF + cosine similarity	NER	Resumes and job descriptions	Matching, transparency, privacy
6	Optimizing job matching [6]	SVM	LR, RF, KNN, TF-IDF	Resume/job-role data	Classification performance
7	NLP-powered resume matching [7]	TF-IDF + cosine similarity	PDF extraction, ML fit model	PDF resumes and job descriptions	Semantic relevance and job fit
8	LLM-driven resume matching [8]	GPT-based models	Fine-tuning, RAG, prompt engineering	998 real resumes; 15 job areas	Processing time and fit assessment
9	Chandak et al. (2024) [9]	Spa Cy + Regex	PDF Miner, TF-IDF, cosine similarity	PDF resumes; job-title/job-listing dataset	Skill extraction and job recommendation; no overall parser accuracy stated
10	Jayakumar et al. (2023) [10]	AI recruitment framework	Resume parser, candidate screening, recommendation, HR analytics	Applicant resumes; employee-attrition dataset	Top-three candidate matches >80%; attrition model metrics are a separate task
11	Cap Style BERT (2024) [11]	Style-aware BERT	Capitalization, font size, bold/italic features; domain pre-training	7,219 resumes; ~2M pre-training resumes	Resume segmentation F1 improves ~3–6%; +5.4% with 5% training data
12	Gaur et al. (2021) [12]	Semi-supervised deep NER	PDF2JSON, NLTK POS tagging, correction and retraining	550 seed resumes; education sections	92.06% accuracy; 71.13% precision; 75.56% recall; 73.28% F1
13	Wos I a k (2021) [13]	Hybrid NER + dictionaries	Liner2, NERF, Babel f y, CRF, fuzzy matching, section splitting	1,326 structured CVs; 17,000 CVs for CRF tests	Institutions: 30.92%; companies: 19.68% detection by hybrid parser
14	War us a with ana et al. (2023) [14]	Layout-aware SVM classifier	Word-box spacing, TF-IDF, SGD Classifier, custom NER, regex	PDF resumes with section annotations	SVM accuracy 0.888; Naive Bayes 0.840; section F1 varies
15	Arokiaraj et al. (2024) [15]	Gemini Pro Vision LLM	Stream lit, PDF-to-image, prompt-based matching and feedback	Single/multiple resumes and job descriptions	Strengths, weaknesses, missing skills and match percentage; no benchmark metrics stated
16	Tejaswini et al. (2022) [16]	TF-IDF + cosine + KNN	Content-based recommendation and text preprocessing	CVs and supplied job descriptions	85% average text-parsing accuracy; 92% ranking accuracy

17	Proposed Resume Parser AI	Skill clustering + weighted score	NLTK, preprocessing, skill matching	Kaggle Resume Dataset (planned)	Precision, recall, F1, ranking metrics
18	Hybrid screening approach	NER + semantic matching	TF-IDF, cosine similarity, human review	Domain-specific labelled resumes	Robustness, fairness, explainability

V. PROPOSED SYSTEM OVERVIEW

The proposed Resume Parser AI follows a sequence of document intake, text extraction, preprocessing, skill identification, skill grouping, job matching, and score generation. First, the resume text and target job description are collected. Text is then normalized by converting it to a consistent case, removing unnecessary punctuation and extra spaces, tokenizing sentences and words, and filtering common stop words. Stemming or lemmatization may be applied where it improves consistency. NLTK provides tools for several of these preprocessing operations.

After preprocessing, the system identifies relevant information such as technical skills, tools, qualifications, projects, and work experience. A skill dictionary or labelled examples can support the initial extraction process. Related terms can then be grouped using a clustering method so that closely related words and skills are treated as belonging to a common job domain. The selected clustering algorithm and number of clusters should be determined through validation on the available dataset.

The candidate profile is compared with the job requirements. A baseline implementation may represent resume and job-description text using TF-IDF vectors and calculate cosine similarity. A skill-coverage component can separately measure how many required skills are evidenced in the resume. Experience and project relevance can be included when the source text provides sufficient information. Repeated keyword occurrences should not automatically increase the score; evidence, context, and distinct skill coverage should be considered.

The final relevance score is normalized to the range 0–10. The system can also display matched skills, partially matched skills, missing mandatory skills, and the text sections supporting each match. This makes the result easier for a recruiter to verify. The ranking is intended to prioritize manual review and should not independently decide employment outcomes.

VI. CHALLENGES AND FUTURE SCOPE

Resume documents vary in file format, layout, language, section naming, and writing style. PDF text extraction may fail for scanned documents or complex columns, while abbreviations and spelling variations can reduce skill recognition. A dataset containing resume categories does not necessarily provide job-specific relevance labels; therefore, a separate set ofresume–job-description pairs should be labelled for evaluating ranking quality. Another challenge is keyword inflation, where a candidate repeats popular terms without demonstrating practical experience. Context-aware extraction can reduce this problem by checking whether a skill appears in a project, employment description, certification, or skills section. However, automated systems can still make mistakes and must provide understandable evidence for their outputs.

Future work may compare NLTK-based preprocessing with transformer-based embeddings, evaluate different clustering algorithms, and introduce domain-specific skill taxonomies. A larger and more diverse labelled dataset can support reliable evaluation. Fairness audits, privacy safeguards, human review, and transparent score explanations should be included before any real recruitment deployment.

VII. CONCLUSION

This survey reviewed AI and NLP techniques used for resume parsing, skill extraction, job matching, and automated candidate ranking. Existing studies demonstrate the usefulness of text preprocessing, named entity recognition, TF-IDF, cosine similarity, classification algorithms, nearest-neigh bour methods, and semantic representations. The proposed Resume Parser AI combines Python and NLTK preprocessing with skill grouping through clustering and a normalized 0–10 relevance score to support job-specific screening. The comparative review suggests that a hybrid approach can balance computational simplicity, contextual matching, and explainability. Its effectiveness must be validated using labelled resume–job-description pairs and appropriate precision, recall, F1-score, and ranking metrics. The system should remain a recruiter-support tool, with human judgment retained for final selection.

REFERENCES

[1] K. Tejaswini, V. Umadevi, S. M. K ad I w al, and S. Rev anna, "Design and development of machine learning based resume ranking system," Global Transitions Proceedings, vol. 3, no. 2, pp. 371–375, Nov 2022,doi: 10.1016/j.gltp.2021.10.002.

[2] A. Barducci, S. Iannaccone, V. La Gatta, V. Moscato, G. Sper l ì, and S. Za vota, "An end-to-end framework for information extraction from Italian resumes," Expert Systems with Applications, vol. 210, Art. no. 118487, 2022,doi: 10.1016/j.eswa.2022.118487.

[3] "Automated Resume Parsing: A Natural Language Processing Approach," in Proc. 7th Int. Conf. Computation System and Information Technology for Sustainable Solutions (CSITSS), 2023,doi: 10.1109/CSITSS60515.2023.10334236.

[4] "Job Vacancy Ranking with Sentence Embeddings, Keywords, and Named Entities," Information, vol. 14, no. 8, Art. no. 468, 2023.

[5] "Resume Ranking with TF-IDF, Cosine Similarity and Named Entity Recognition," in Proc. First Int. Conf. Data, Computation and Communication (ICDCC), 2024,doi: 10.1109/ICDCC62744.2024.10961659.

[6] "Optimizing Job Matching Through Automated Resume Screening with NLP and Machine Learning," in Proc. IEEE Int. Conf. on Intelligent Signal Processing and Effective Communication Technologies (INSPECT), 2024,doi: 10.1109/INSPECT63485.2024.10896014.

[7] I. Rathi, P. Kola skar, L. Ta g alu, and M. Mali, "NLP-Powered Resume Matching For Recruitment," International Journal for Multidisciplinary Research, vol. 6, no. 6, 2024,doi: 10.36948/ijfmr.2024.v06i06.31742.

[8] "LLM-Driven Resume Parsing and Job Description Matching: Evaluation on 998 Real Resumes," in Proc. IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC), 2026,doi: 10.1109/CCWC67433.2026.11393696.

[9] A. V. Chandak, H. Pandey, G. Ru s hi ya, and H. Sharma, "Resume Parser and Job Recommendation System using Machine Learning," in Proc. 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), 2024,doi: 10.1109/ESIC60604.2024.10481635.

[10] N. Jayakumar, A. K. Maheshwaran, P. S. Arvind, and G. Vijayaragavan, "On-Demand Job-Based Recruitment for Organ sati ons Using Artificial Intelligence," in Proc. 2023 International Conference on Networking and Communications (ICNWC), 2023,doi: 10.1109/ICNWC57852.2023.10127551.

[11] M. Y. Bocharova and E. V. Malakhov, "Cap Style BERT: Incorporating Capitalization and Style Information into BERT for Enhanced Resumes Parsing," in Proc. 13th International Conference on Software and Computer Applications (ICSCA), 2024,doi: 10.1145/3651781.3651820.

[12] B. Gaur, G. S. Saluja, H. B. Sivakumar, and S. Singh, "Semi-supervised deep learning based named entity recognition model to parse education section of resumes," Neural Computing and Applications, vol. 33, pp. 5705–5718, 2021,doi: 10.1007/s00521-020-05351-2.

[13] A. Wos a k, "Automated extraction of information from Polish resume documents in the IT recruitment process," Procedia Computer Science, vol. 192, pp. 2432–2439, 2021,doi: 10.1016/j.procs.2021.09.012.

[14] S. P. W a rusa with ana, N. N. Perera, R. L. Weerasinghe, T. M. Hind akarl den iya, and G. U. Gane go da, "Layout Aware Resume Parsing Using NLP and Rule-based Techniques," in Proc. 8th International Conference on Information Technology Research (ICITR), 2023,doi: 10.1109/ICITR61062.2023.10382773.

[15] A. S. Arokiaraj, T. Amudha, and S. Ramakrishnan, "Applicant Resume Tracking System (ARTS): GenAI Agent," in Proc. ICIMMI 2024, Jaipur, India, 2024,doi: 10.1145/3745812.3745851.

[16] K. Tejaswini, V. Umadevi, S. M. K ad I w a l, and S. Rev anna, "Design and development of machine learning based resume ranking system," Global Transitions Proceedings, vol. 3, no. 2, pp. 371–375, 2022,doi: 10.1016/j.gltp.2021.10.002.

Note: Metrics are quoted in the context of each paper's own task and dataset. In particular, employee-attrition metrics in [10] are not resume-parsing metrics. Papers that do not report a numerical parser benchmark are identified as such.