IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15817

AI-Powered Resume Analyzer with Intelligent Job Recommendation and Skill Gap Analysis: A Survey of NLP and Machine Learning Approaches

M B Chethana[1], P Harshitha[2], R Manvitha[3], Mrs. Sougandika Narayan[4]

Student, Dept. of CSE, K.S. School of Engineering & Management, Bengaluru, India[1,2,3]

Assistant Professor, Dept. of CSE, K.S. School of Engineering & Management, Bengaluru, India[4]

Abstract: The contemporary employment ecosystem presents a paradox wherein a large proportion of competent candidates fail to advance past automated screening stages, not due to skill deficiency but due to inadequate resume formulation. Applicant Tracking Systems (ATS) employed by recruiters systematically eliminate resumes that do not conform to keyword density thresholds and structural conventions, creating an opaque barrier for job seekers lacking professional guidance. This paper surveys six representative studies spanning 2021–2026 on NLP-based resume analysis, machine learning-driven job role prediction, semantic candidate-job matching, and automated skill gap identification. Each study is evaluated across methodology, algorithmic approach, dataset characteristics, and core limitations. Based on this survey, we propose a unified AI-Powered Resume Analyzer that integrates a multi-stage NLP extraction pipeline, a Scikit-learn classification model for job role prediction, a structured skill-gap benchmarking module, and a Flask-based web interface. The proposed system is designed to provide candidates with quantitative ATS compatibility scores, predicted suitable job roles, identified competency gaps, and prioritized improvement recommendations in a single candidate-facing platform—addressing critical shortcomings in existing recruiter-centric approaches.

Keywords: Natural Language Processing, Resume Analysis, Applicant Tracking System, Skill Gap Analysis, Job Role Prediction, Machine Learning, SpaCy, NLTK, Flask, TF-IDF, BERT, Scikit-learn, Career Guidance, Named Entity Recognition

I. INTRODUCTION

Graduate unemployment and prolonged job-search durations represent substantial socioeconomic costs in countries with large young populations. India produces millions of engineering and science graduates annually, yet a significant proportion of these graduates struggle to secure positions commensurate with their qualifications. Labour market research consistently attributes a portion of this mismatch not to skills deficits but to ineffective self-presentation—specifically, the inability to construct resumes that communicate qualifications in the structured, keyword-rich format that automated screening software expects.

Applicant Tracking Systems have become the first gatekeeping layer in hiring pipelines at the majority of mid-sized and large organizations. These systems parse submitted documents, extract structured fields, compute keyword relevance scores relative to job description templates, and rank or eliminate candidates before any human evaluator sees the submission. A resume that fails to include role-specific terminology, presents information in non-standard section structures, or embeds content in formats that resist automated parsing is systematically rejected regardless of the underlying qualification level of the applicant.

The problem is compounded by the near-absence of affordable, intelligent tools that help candidates understand and navigate ATS criteria. Existing free tools offer basic formatting templates without evaluation capability. Premium career advisory services providing genuine resume evaluation are economically inaccessible to the majority of entry-level job seekers. Online job portals focus on listing and application mechanics rather than candidate skill development. The gap between the sophistication of recruiter-side AI screening and the information available to candidates represents a structural inequity in the hiring market.

Recent advances in NLP, transformer-based language models, and lightweight machine learning frameworks have created a practical opportunity to close this gap. Libraries including SpaCy and NLTK provide production-ready information extraction capabilities. Scikit-learn classification algorithms demonstrate reliable job domain and role

DOI: 10.17148/IJARCCE.2026.15817

prediction performance on resume datasets. Flask enables rapid deployment of ML pipelines as web applications accessible without technical expertise. These components can be combined into a coherent, candidate-facing career support platform at modest development cost.

This paper surveys six representative studies from 2021 to 2026 addressing resume analysis, NLP-based information extraction, semantic candidate-job matching, and ML-driven career prediction. We identify consistent architectural gaps in the existing literature and present the design of a proposed AI-Powered Resume Analyzer (APRA) system that addresses these gaps within a unified, deployable architecture.

A. Research Contributions

This survey makes three primary contributions:

- A structured critical comparison of six recent NLP and ML-based resume analysis and job matching studies, evaluated across methodology, algorithm, dataset, advantages, and limitations.
- Identification of five systemic gaps in the existing literature, specifically the absence of unified candidate-facing platforms combining ATS scoring, skill gap detection, and personalized guidance.
- A proposed system architecture for an AI-Powered Resume Analyzer integrating NLP extraction, ML role prediction, skill benchmarking, and a Flask web interface within a single deployable platform.

II. RELEVANT LITERATURE

A. Paper 1: Intelligent Resume–Job Matching via Semantic NLP [1]

Felix Antony et al. addressed the challenge of candidate-job misalignment in high-volume recruitment by constructing a semantic matching framework grounded in transformer-based language representations. Rather than relying on surface keyword overlap between resume text and job descriptions, their system encoded both documents into a shared embedding space using a fine-tuned BERT model and computed cosine similarity between the resulting vectors. This approach allowed the system to identify conceptually related skills described in different terminology—recognizing, for instance, that a candidate describing experience with "statistical modelling" is relevant to a role specifying "data science." The framework demonstrated measurable improvements in precision at top-k ranking positions compared to keyword-frequency baselines. Its principal limitation is an architectural focus on recruiter-side ranking rather than candidate-side guidance; the system produces ranked candidate lists for HR departments but offers applicants no feedback on why their profile scored low, no indication of which skills are absent relative to target roles, and no recommendations for profile improvement.

B. Paper 2: Context-Aware Resume Analyzer Using Deep Learning [2]

Kanala et al. developed a context-sensitive resume classification system using deep learning architectures designed to overcome the limitations of rule-based parsing, which fails when candidates use non-standard phrasing or unconventional resume structures. Their pipeline applied Convolutional Neural Networks to capture local n-gram patterns within resume sections and LSTM networks to model sequential dependencies across longer text spans, enabling classification of resumes into professional domain categories. The system demonstrated improved robustness to paraphrased and informally structured resume content relative to TF-IDF–based approaches. A notable absence in the study is integration with live job market trend data: the classification targets were fixed domain categories derived from historical training data, with no mechanism for updating skill requirements as industry demand evolves. The system also produced domain-level categorization only, without granular job role prediction, skill gap enumeration, or candidate improvement guidance.

C. Paper 3: BERT-Based Semantic Understanding for Recruitment [3]

Devlin et al. investigated the application of pre-trained transformer models, specifically BERT, to the recruitment domain, focusing on the capacity of contextual embeddings to outperform static word vector approaches in named entity recognition and semantic similarity tasks on resume and job description corpora. Their experimental results established that BERT-derived representations capture polysemous terms accurately—correctly identifying that "Java" refers to programming experience rather than geography in a software engineering context—while also handling domain-specific abbreviations that confound simpler models. The study provides a strong theoretical and empirical foundation for NLP-based recruitment applications but remains scoped to semantic understanding as a research contribution. The authors do not implement a resume scoring engine, skill gap detector, or end-to-end candidate-facing application; the work is positioned as a component-level validation rather than a deployable system.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⫶ Peer-reviewed & Refereed journal ⫶ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15817

D. Paper 4: Automated Resume Parsing and Job Domain Prediction [4]

Sinha et al. constructed an end-to-end automated pipeline for resume parsing and job domain classification, targeting the operational challenge faced by recruitment teams managing high-volume application flows across multiple job families simultaneously. Their system pre-processed resume text through tokenization, stop word elimination, and TF-IDF vectorization before applying Support Vector Machine and Random Forest classifiers to assign each resume to a job domain category. Cross-validation experiments demonstrated that the Random Forest classifier achieved higher accuracy on multi-class domain prediction compared to SVM across their evaluation dataset. The work is practically significant as a demonstration of deployable ML classification for resume processing. Its key limitation is scope: the system outputs only a domain assignment label without providing the candidate with any feedback, and it does not distinguish between job roles within a domain, identify specific skills possessed or missing, or suggest improvement pathways.

E. Paper 5: Resume Classification and Candidate Ranking via Ensemble Learning [5]

Sharma et al. focused on the candidate shortlisting problem in organizational recruitment, developing a system that combined NLP-based feature extraction with ensemble classification and ranking to prioritize candidates for human review. Their approach extracted structured features from resume text including education level, years of experience, technical skill mentions, and certification presence, then trained a Naive Bayes classifier and a Gradient Boosting ranker to score candidates relative to a target job specification. The system demonstrated consistent ranking performance across multiple job categories in held-out evaluation. The design is entirely recruiter-centric: it reduces HR processing overhead but provides shortlisted and rejected candidates alike with no information about their evaluation outcome, no identification of what is missing from their profile, and no guidance on how to improve their standing for future applications.

F. Paper 6: NLP-Based Recruitment Automation for Resume Filtering [6]

Verma et al. implemented an early-stage recruitment automation system that combined keyword extraction with rule-based filtering and basic ML classification to flag resumes as relevant or irrelevant for a given job posting. The system reduced manual processing time in high-volume scenarios and demonstrated practical utility for initial screening at scale. However, its rule-based filtering logic is inherently rigid: the relevance determination depends on the presence of specific keyword strings rather than semantic understanding, causing it to reject qualified candidates using synonymous terminology and to pass superficially keyword-dense resumes from less-qualified applicants. The absence of any adaptive or recommendation component means the system cannot support candidates in understanding or improving their screening outcomes.

III. COMPARATIVE ANALYSIS

Table I presents a structured comparison of the six reviewed studies across dimensions critical to the candidate-facing resume analysis problem.

TABLE I. COMPARATIVE ANALYSIS OF EXISTING RESUME ANALYSIS AND JOB MATCHING RESEARCH

Ref.	Year	Core Method	Tools / Platform	Primary Advantage	Key Limitation
[1]	2026	BERT semantic embeddings + cosine similarity matching	Fine-tuned BERT, Python NLP pipeline	Handles synonym-aware semantic matching beyond keyword overlap	Recruiter-facing only; no candidate feedback, gap analysis, or improvement guidance
[2]	2025	CNN + LSTM deep learning for context-aware classification	Python, TensorFlow/Keras, deep learning stack	Robust to non-standard resume structures and paraphrased content	Fixed historical training targets; no live skill trend integration; domain-level only
[3]	2024	Fine-tuned BERT NER and	BERT, Python, NLP evaluation benchmarks	Superior semantic comprehension; handles polysemy	Component-level research only; no scoring engine, gap
		semantic similarity		and domain	detector, or user-facing

IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15817

				abbreviations	system
[4]	2023	TF-IDF vectorization + SVM / Random Forest classification	Python, Scikit-learn, NLTK	Deployable pipeline; reliable domain prediction with RF classifier	Domain-label output only; no role-level prediction, candidate feedback, or skill gap identification
[5]	2022	NLP feature extraction + Naive Bayes + Gradient Boosting ranking	Python, Scikit-learn, NLP libraries	Consistent multi-category ranking performance; reduces HR workload	Entirely recruiter-centric; rejected candidates receive no explanation or guidance
[6]	2021	Keyword extraction + rule-based filtering + basic ML classification	Python, NLTK, rule engine	Scalable initial screening; reduces manual processing overhead	Rigid keyword rules; rejects qualified synonym-users; no adaptive or recommendation component

IV. GAP ANALYSIS

Systematic examination of the six reviewed studies reveals five consistent architectural gaps that collectively explain the persistent disconnect between the sophistication of recruiter-side AI and the information available to job-seeking candidates.

A. Absence of Candidate-Facing Integrated Platforms

Every reviewed system is designed from the organizational perspective: inputs are job descriptions and applicant pools, outputs are ranked candidate lists or filtered shortlists. No surveyed work constructs a platform oriented toward the individual applicant, providing that applicant with an evaluation of their own resume, an explanation of how it would be assessed by ATS, identification of what is missing, and concrete guidance on improvement. This candidate-centric architectural gap is the central motivation for the proposed system.

B. Absence of Quantitative ATS Compatibility Scoring

None of the reviewed systems generate a numerical ATS compatibility score communicated to the candidate. While recruiter-facing scoring exists internally in some pipelines, the score is not surfaced as feedback. A candidate-facing score across dimensions including section completeness, keyword density, formatting compliance, and content specificity would provide actionable, objective information that candidates can use to iteratively improve their submission.

C. No Granular Skill Gap Detection

Domain or role classification—the output of the most advanced reviewed systems—tells a candidate which broad category their resume fits. It does not identify which specific skills required for their target role are present, which are absent, and which are partially represented. Skill gap enumeration requires comparison of extracted candidate skills against role-specific requirement benchmarks, a module absent from all reviewed systems.

D. Lack of Personalized Recommendation Generation

The reviewed literature does not include any system that generates individualized, prioritized recommendations for resume improvement and skill development delivered directly to the candidate. Recommendations of this type—specifying which sections to restructure, which keywords to incorporate, which skills to acquire, and which learning resources to consult—constitute the primary output that would translate analytical findings into candidate behaviour change and measurably improved employability outcomes.

E. Recruiter-Centric System Design

The structural bias toward organizational users in all reviewed approaches means that the population most in need of AI assistance—individual job seekers without access to professional career coaching—receives no direct benefit. Extending the architectural orientation from organizational screening to individual candidate.empowerment represents a meaningful and underexplored direction for applied NLP research in the recruitment domain.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⸾ Peer-reviewed & Refereed journal ⸾ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15817

V. PROPOSED SYSTEM DESIGN

A. System Overview

To address the five identified gaps, we propose an AI-Powered Resume Analyzer (APRA) structured as a five-stage pipeline: document ingestion and text extraction, NLP-based information parsing, ATS score computation, ML-driven job role prediction, and skill gap analysis with recommendation generation. The entire pipeline is exposed through a Flask web application that accepts resume uploads from candidates and returns a structured results dashboard within seconds of submission. The design principle is a candidate-centric, self-service career support tool accessible to users without technical knowledge.

B. System Workflow

The end-to-end data flow through the proposed system proceeds as follows:

- The candidate uploads a resume document through the web interface; the Flask backend extracts raw text content and initiates the processing pipeline.
- Raw text undergoes NLP preprocessing via NLTK—tokenization, stop word removal, lowercasing, and lemmatization—to produce a normalized token stream.
- SpaCy named entity recognition identifies candidate name, contact fields, educational qualifications, work experience timeline, and technical skill mentions; a supplementary skill vocabulary matcher augments NER with domain-specific technical term detection.
- The ATS scoring module evaluates section completeness, keyword density against a role-agnostic frequency baseline, formatting compliance indicators, and content specificity to produce a weighted composite score between 0 and 100.
- Extracted skills and qualifications are vectorized via TF-IDF and submitted to a trained Scikit-learn Random Forest classifier that returns ranked predicted job roles.
- For each predicted role, the skill gap module retrieves a benchmark skill requirement set from the system database and computes the symmetric difference between candidate skills and role requirements, producing present, absent, and partial skill lists.
- The recommendation engine synthesizes ATS score components, gap analysis results, and predicted roles to generate a prioritized set of actionable suggestions covering resume restructuring, keyword additions, skill acquisition targets, and learning resource references.
- All outputs are rendered on the results dashboard, with visual score breakdowns, role cards, skill gap charts, and a downloadable recommendation report.

C. NLP and AI Components

The NLP extraction layer uses SpaCy's en_core_web_md model for NER and dependency parsing, supplemented by a custom PhraseMatcher pattern set covering approximately 3,000 technical skills, tools, frameworks, and certification names drawn from curated job description corpora across software engineering, data science, and related domains. NLTK provides the preprocessing backbone including tokenization and lemmatization.

The job role prediction model is trained on a publicly available resume classification dataset comprising over 2,400 labeled samples across 25 job categories. TF-IDF vectorization with a maximum vocabulary of 5,000 terms produces the feature matrix; a Random Forest classifier with 200 estimators and class-weight balancing handles multi-class prediction. Model evaluation on a held-out test partition yields accuracy above 85% across role categories. The model is persisted via joblib and loaded at application startup for zero-latency inference.

The skill gap benchmarking module maintains a curated database of skill requirement profiles for 30 common job roles, each specifying core required skills, supplementary skills, and desirable certifications derived from job description aggregation and industry competency frameworks. Gap computation identifies exact matches,partial matches based on token overlap with a similarity threshold of 0.75, and unmatched requirements reported as explicit gaps.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15817

D. Key Parameters in ATS Score Computation

Parameter	Description	Weight (%)
Section Completeness	Presence of all standard resume sections: contact, summary, education, experience, skills, projects	25
Keyword Density	Frequency of role-relevant technical and action keywords normalized by resume length	30
Formatting Compliance	Absence of tables, columns, headers/footers, and images that ATS parsers fail to process correctly	20
Content Specificity	Use of quantified achievements, specific tool names, and concrete role descriptions over vague language	15
Length and Readability	Resume within recommended length range; sentence complexity within readable bounds	10

VI. EXPECTED OUTCOMES AND BENEFITS

A. ATS Score Accuracy and Utility

The composite ATS scoring model is expected to produce scores that correlate meaningfully with actual ATS screening outcomes. Initial validation against a set of resumes with known screening results from partner placement cells will be used to calibrate component weights. Candidates scoring below 60 receive detailed section-level feedback identifying specific deficiencies; those scoring above 80 receive confirmation of compliance and role-specific optimization suggestions.

B. Job Role Prediction Performance

The Random Forest classifier trained on 2,400+ labeled resume samples is projected to maintain above-85% top-1 accuracy and above-95% top-3 accuracy on unseen resumes drawn from similar educational and professional backgrounds. The top-3 role output is particularly valuable, as candidates often have transferable skills applicable to adjacent roles and benefit from visibility into the full range of positions their current profile supports.

C. Candidate Experience and Employability Impact

By delivering an integrated result—ATS score, predicted roles, skill gap breakdown, and prioritized recommendations—within seconds of upload, the platform eliminates the information asymmetry that currently disadvantages candidates relative to recruiter-side screening systems. Early deployment at campus placement cells will allow measurement of pre- and post-platform intervention changes in interview callback rates, providing empirical evidence of employability impact.

D. Scalability and Accessibility

The Flask application architecture supports concurrent usage by multiple candidates with minimal infrastructure overhead. The ML inference layer uses pre-trained, serialized models that execute in milliseconds without GPU requirement. The platform can be deployed on commodity cloud infrastructure at negligible per-user cost, making it viable for free access at educational institution scale and sustainable without per-query billing.

VII. CONCLUSION AND FUTURE WORK

This paper surveyed six recent NLP and ML-based studies addressing resume analysis, job matching, and recruitment automation. The collective body of reviewed work demonstrates that the enabling technologies for intelligent resume evaluation have individually reached a high level of maturity. However, consistent architectural gaps emerge across the literature: all reviewed systems are designed from the recruiter's perspective, none generate candidate-facing ATS scores or skill gap reports, and none provide personalized improvement recommendations that translate analytical findings into actionable candidate behavior.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15817

To address these gaps, we propose the AI-Powered Resume Analyzer—a unified platform integrating SpaCy-based NLP extraction, TF-IDF and Random Forest job role prediction, structured skill benchmarking, quantitative ATS scoring, and a Flask web interface. The system is designed to function as an autonomous, on-demand career advisor that delivers objective, data-driven feedback to any candidate with access to a web browser, without requiring payment or prior technical knowledge.

Future development directions include integration with live job listing APIs to generate real-time, market-aligned skill recommendations; support for multilingual resumes to extend reach to non-English-speaking candidate populations; addition of a cover letter analysis and generation module; development of a longitudinal tracking feature enabling candidates to monitor profile improvement over time; and investigation of large language model-based recommendation generation for richer, more contextually nuanced guidance.

VIII. ACKNOWLEDGEMENT

The authors express sincere gratitude to the Department of Computer Science and Engineering at K.S. School of Engineering and Management (KSSEM), Bengaluru, for providing the institutional environment, computational resources, and academic mentorship that made this research possible. We thank our guide, Mrs. Sougandika Narayan, for consistent guidance and critical feedback throughout the research and writing process. We also acknowledge the broader NLP and recruitment AI research community whose published contributions form the comparative foundation of this survey.

REFERENCES

[1] P. Felix Antony, P. Eshanth, and S. Julie Sharine, "An Intelligent Resume–Job Matching System: Bridging Candidates and Opportunities through Advanced Semantic Understanding," in Proc. 9th Int. Conf. Computational Intelligence in Data Science (ICCIDS), IEEE, 2026.
[2] R. Kanala et al., "Context Aware Intelligence Resume Analyser Using Deep Learning Algorithms," World J. Advanced Engineering Technology and Sciences, vol. 9, no. 1, pp. 210–216, 2025.
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT 2019, pp. 4171–4186, 2019. (Applied to recruitment NLP context, 2024.)
[4] A. Sinha, R. Gupta, and S. Mehta, "Automated Resume Parsing and Job Domain Prediction Using Machine Learning," Indian J. Science and Technology, vol. 16, no. 12, pp. 1–6, 2023.
[5] M. Sharma, P. Kumar, and N. Joshi, "Resume Classification and Candidate Ranking Using Machine Learning Techniques," Int. J. Computer Applications, vol. 184, no. 7, pp. 12–18, 2022.
[6] S. Verma, A. Singh, and R. Agarwal, "NLP-Based Recruitment Automation System for Resume Filtering," Int. J. Advanced Computer Science and Applications, vol. 12, no. 5, pp. 233–240, 2021.