ISSN: 2455 - 1597

ijergs
Open Access

International Journal of Engineering Research and Generic Science (IJERGS)
Available online at: https://www.ijergs.in

Volume - 12, Issue - 3, May – June – 2026, Page No. 124 – 130

Smart Placement Management System with Resume Analyzer

[1]Mr. Ankit Rai, Department of MCA IIMT College of Engineering, Greater Noida

[2]Ms. Janhvi Srivastava, Department of MCA IIMT College of Engineering, Greater Noida

[3]Ms. Shweta Kumari, Department of MCA IIMT College of Engineering, Greater Noida

[4]Dr. Naveen Kumar Sharma, Department of MCA IIMT College of Engineering, Greater Noida

Abstract

The Smart Placement Management System with Resume Analyzer is an integrated AI-powered platform designed to streamline and automate the campus placement process for educational institutions. Traditional placement processes involve manual resume screening, inefficient job-candidate matching, and limited analytics, leading to poor placement outcomes. This paper presents a comprehensive system that leverages Natural Language Processing (NLP), Machine Learning (ML), and rule-based algorithms to automate resume parsing, candidate ranking, skill gap analysis, and job role matching. The proposed system reduces placement cycle time by approximately 60%, improves candidate-to-role match accuracy to 87.4%, and provides actionable analytics to both students and placement officers. Experimental evaluation on a dataset of 500 student profiles and 120 job descriptions demonstrates the effectiveness of the hybrid AI approach over conventional methods.

Keywords: Placement Management System, Resume Analyzer, NLP, Machine Learning, Candidate Matching, Skill Gap Analysis, Campus Recruitment Automation.

Introduction

Campus placement is a critical milestone for students in higher education institutions. It bridges the gap between academic learning and industry requirements. However, most institutions continue to rely on manual, paper-based placement processes that are time-consuming, error-prone, and fail to leverage the vast amounts of candidate and job data available today. The rapid growth of artificial intelligence and natural language processing has opened new avenues for automating knowledge-intensive tasks. Resume screening — one of the most laborintensive activities in the placement process — is particularly amenable to AI-based automation. Industry surveys indicate that human recruiters spend less than10 seconds reviewing a single resume, leading to significant cognitive bias and high rates of qualified candidate rejection.

The Smart Placement Management System (SPMS) with Resume Analyzer addresses these challenges by providing a unified platform where:

Students can upload resumes and receive instant AI-driven feedback and skill gap analysis.

Placement officers can manage job postings, review ranked candidate lists, and generate placement reports. Companies can post opportunities, define skill requirements, and receive AI-curated shortlists. The system integrates rule-based resume parsing with ML-based job matching and a recommendation engine, ensuring both precision and adaptability. This paper describes the system architecture, component design, experimental evaluation, and future research directions.

Literature Review

Significant research has been conducted in the domain of automated resume analysis and intelligent recruitment systems.

- **Resume Parsing and Information Extraction** Early approaches to resume parsing relied on template-based and rulebased methods, which performed well only for standardized resume formats. Subsequent work applied Named Entity Recognition (NER) using Conditional Random Fields (CRFs) to extract structured information such as name, education, skills, and experience.

  More recently, transformer-based models like BERT have demonstrated superior performance in extracting nuanced resume sections with accuracy exceeding 91% on benchmark datasets [4].

- **Job-Candidate Matching**

  Job-candidate matching has been formulated as a text similarity problem, a classification problem, and a recommendation problem. Vector space models using TF-IDF and cosine similarity were among the first methods applied. Graph-based approaches that model the relationship between skills, roles, and candidates have shown improvement in multi-dimensional matching scenarios. Collaborative filtering techniques analogous to those used in e-commerce recommendation systems have also been successfully adapted for resume-to-job matching.

- **Skill Gap Analysis**

  Skill gap analysis in automated systems typically involves comparing extracted candidate skills against role-specificontologies or taxonomies. The ESCO (European Skills, Competences, Qualifications and Occupations) framework provides a structured vocabulary used in several recent systems Deep learning approaches have demonstrated the ability to infer implicit skills from experience descriptions using word embedding techniques.

- **Placement Management Systems**

  Existing placement management systems such as Campus, Simplicity, and Handshake primarily focus on administrative workflow management and lack intelligent analytics. Research prototypes have demonstrated the value of integrating predictive analytics into placement systems for forecasting placement probability and identifying at-risk students. However, a comprehensive system integrating resume intelligence, smart matching, and placement analytics remains an open research challenge that this work addresses.

System Architecture

The SPMS is designed as a three-tier web application with a micro services-based AI processing layer. Figure 1 illustrates the overall architecture.

- **High-Level Architecture**

The system comprises five primary modules:

Resume Intake and Parser Module: Accepts PDF/DOCX resumes, extracts structured information.

- Skill Extraction and Gap Analysis Engine: Maps extracted skills to acurated ontology and identifies missing competencies.
- Job Description Processor: Parses company-posted job descriptions and generates structured requirement vectors.

- Candidate Ranking and Matching Engine: Computes multidimensional similarity scores and generates ranked shortlists. Analytics and Reporting Dashboard: Provides real-time placement statistics, trend analysis, and predictive insights.

➢ **Technology Stack**

The backhand is implemented in Python (Flask/Fast-API) with MongoDB for document storage and PostgreSQL for relational data. The NLP pipeline uses spacey and Hugging-face Transformers. The front-end is built with React.js, and the system is containerized using Docker with deployment on cloud infrastructure.

Methodology

➢ **Resume Parsing Pipeline**

The resume parsing pipeline operates in five sequential stages. First, document prepossessing converts uploaded files to plain text using Apache Tikka, handling PDF, DOCX, and DOC formats. Second, sentence segmentation and ionization are performed using spacer's en_core_web_lg model. Third, a custom NER model fine-tuned on a corpus of 2,000 annotated resumes extracts entities including PERSON, ORG, DATE, DEGREE, SKILL, and CGPA. Fourth, section classification assigns parsed content to canonical sections (Education, Experience, Skills, Projects, Certifications) using a fine-tuned BERT classifier with 93.7% accuracy on the validation set. Fifth, structured data is serialized to a canonical JSON schema for downstream processing.

➢ **Skill Extraction and Ontology Mapping**

Skills extracted by the NER model are normalized against a curated skill ontology comprising 4,800 technical and soft skills organized in a three-level hierarchy: Domain → Category → Skill. The ontology was built by aggregating data from ESCO, O*NET, and Linked-in's standardized skill taxonomy.

Fuzzy string matching using the Jar

Winkles algorithm resolves orthographic variants (e.g., 'ML', 'machine learning', 'Machine Learning'). Synonym resolution is performed using a skill embedding space trained on job postings corpus using fast-text.

➢ **Job Description Processing**

Job descriptions submitted by recruiters are processed through an analogous pipeline. Required and preferred skills are extracted separately to enable weighted matching. Experience level, educational qualification requirements, and salary range are extracted using regex patterns and ML classifiers. Each processed job description is represented as a structured requirement vector in the same ontology space as candidate profiles.

➢ **Candidate Ranking and Matching Algorithm**

The matching algorithm computes a composite score S(c, j) for candidate c and job j as:

$S(c, j) = \alpha \cdot S\ skill(c, j) + \beta \cdot S\ exp(c, j) + \gamma \cdot Seduce(c, j) + \delta \cdot S\ project(c, j)$ where S skill is the weighted skill match ratio, S exp is experience relevance score, Seduce is education qualification match, and S project is project domain relevance. Weights α, β, γ, δ are learned via logistic regression trained on historical placement outcomes (α=0.45, β=0.25, γ=0.20, δ=0.10). The final candidate list is ranked by S(c, j) in descending order.

- **Skill Gap Analysis**

For each candidate-job pair, the system generates a skill gap report listing required skills absent from the candidate's profile. Skills are categorized as Critical (required skills with high job weight), Important (required skills with medium weight), and Optional (preferred but not required). The system additionally recommends online learning resources (MOOC links) for each missing critical skill using a curated content database.

Implementation

- **User Roles and Workflows**

The system supports three primary user roles. Students register and upload resumes, view their parsed profile, review skill gap reports, apply to available positions, and track application status. Placement Officers manage the company database, review AI-generated shortlists, schedule interviews, and access placement analytics. Company Recruiters post job descriptions, define requirements, view ranked candidate profiles, and communicate interview outcomes.

- **Resume Feedback Interface**

A key differentiating feature of the SPMS is its real-time resume feedback module. Upon upload, students receive a Resume Strength Score (RSS) on a scale of 0–100, computed from completeness, formatting quality, keyword density, quantification of achievements, and ATS (Applicant Tracking System) compatibility. Section-wise feedback is provided with specific improvement suggestions, enabling students to interactively improve their profiles before applying.

- **Placement Analytics Dashboard**

The analytics dashboard provides institutional-level insights including placement rate trends by department and year, most in-demand skills vs. current student competencies, companywise hiring patterns, average package statistics, and predictive placement probability for upcoming students based on their profiles.Experiments were conducted using data collected from a sample institution comprising 500 student profiles (across Computer Science, Electronics, and Mechanical engineering) and 120 job descriptions from 35 companies over two academic placement cycles. It's recommended to insert figures inside a text box. So your figure would go here (adjusting the text box to the appropriate size to match).

Resume Parsing Accuracy

Table 1: shows the entity extraction performance of the resume parser across key entity types.

Entity Type	Precision (%)	Recall (%)	F1-Score (%)
Name / Contact	98.2	97.8	98.0
Educational Qualification	94.5	93.1	93.8
Skills	91.3	89.7	90.5
Work Experience	88.6	87.2	87.9
Projects	85.4	84.0	84.7
Certifications	92.1	90.5	91.3

Candidate Matching Performance

The matching algorithm was evaluated by comparing AI-generated shortlists against historical human recruiter selections. Table 2 summarizes matching performance across different job categories.

Table 2: Candidate Matching Performance by Job Category

Job Category	Precision@5 (%)	Recall@5 (%)	NDCG@10
Software Development	89.3	86.7	0.912
Data Science / ML	87.1	84.5	0.894
Embedded Systems	83.6	81.2	0.871
Management Trainee	80.4	78.9	0.849
Overall Average	87.4	84.8	0.889

System Performance Metrics

Average resume processing time was 3.2 seconds per document. The complete matching computation for a 500candidate pool against a single job description completed in 1.8 seconds on standard cloud hardware (4 vCPU, 8 GB RAM). The system maintained 99.1% uptime over a 6-month pilot deployment.

User Satisfaction

A survey of 120 students and 15 placement officers following the pilot revealed that 84% of students found the resume feedback module 'very useful' or 'extremely useful', and 91% of placement officers reported that AI generated shortlists reduced their manual screening time by more than half. The overall system usability score (SUS) was 78.4 out of 100, indicating good usability.

Discussion

The results demonstrate that the hybrid NLP and ML approach delivers strong performance across all evaluated dimensions. The high precision in resume parsing (F1 > 90% for most entity types) validates the efficacy of combining rule-based patterns with a fine-tuned NER model. The matching algorithm's NDCG@10 of 0.889 compares favorably with state-of-the-art commercial systems reported in the literature.

An important finding is the performance differential across job categories. Software Development roles yielded the highest matching accuracy, likely because these roles have well-defined, enumerable skill requirements. Management Trainee roles showed relatively lower performance, reflecting the challenge of matching soft skills and personality traits — an area that remains a limitation of the current system.

The skill gap analysis module was particularly valued by students, as it transformed the system from a passive matching tool to an active career development platform. This aligns with findings by that students engage more deeply with placement systems when provided personalized, actionable recommendations.

One limitation observed was the cold-start problem for new companies without historical hiring data, which reduces the reliability of initial shortlists for those companies. This is consistent with challenges reported in similar recommendation systems [8] and is addressed in the future scope.

Future Scope

Several promising directions remain for future research and development: Integration of Large Language Models (LL Ms) such as GPT-4 for contextual resume understanding, enabling assessment of project complexity, leadership indicators, and narrative quality beyond keyword matching. Multi-modal resume analysis incorporating Linked-in profiles, GitHub

repositories, and portfolio websites to create a richer, multi-source candidate profile. Bias detection and fairness auditing modules to ensure AI-generated shortlists do not perpetuate demographic or educational biases present in historical data. Conversational AI interface enabling students to interact with the system via natural language to explore career paths, prepare for interviews, and receive coaching. Federated learning architecture to enable collaborative model training across multiple institutions without sharing sensitive student data.

Conclusion

This paper presented the Smart Placement Management System with Resume Analyzer, an AI-powered platform that comprehensively automates campus recruitment workflows. The system integrates NLP based resume parsing, ontology-driven skill extraction, ML-based candidate ranking, and real-time analytics into a unified platform accessible to students, placement officers, and recruiters. Experimental evaluation demonstrated resume parsing F1scores exceeding 90% for most entity types, an overall candidate matching NDCG@10 of 0.889, and a 60% reduction in placement officer workload. Student satisfaction surveys validated the practical utility of the AI-driven resume feedback and skill gap analysis modules. The convergence of NLP and machine learning in placement management systems represents a It's recommended to insert figures inside a text box. So your figure would go here (adjusting the text box to the appropriate size to match). significant advancement over conventional software tools. As AI capabilities continue to evolve, future systems will move toward fully contextual understanding of candidate profiles, enabling truly personalized career guidance integrated with institutional placement workflows. The architecture and methodology presented in this paper provide a solid foundation for such future developments.

References

1. Liff, S., & Shepherd, A. (2004). An Empirical Study of Resume Screening Bias. Journal of Human Resources, 39(4), 824–847.
2. Ciravegna, F., & Lavelli, A. (2004). Learning Pinocchio: Adaptive Information Extraction for Real-World Applications. Natural Language Engineering, 10(2), 145–165.

3. Yu, K., Guam, G., & Zhou, M. (2005). Resume Information Extraction with Cascaded Hybrid Model. In Proceedings of the ACL.
4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pres-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
5. Siting, Z., Weeing, H., Zhang, N., & Fan, Y. (2012). Job Recommender Systems: A Survey. ICSIT.
6. Salton, G., & Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. Information Processing & Management, 24(5), 513–523.
7. Li, X., Liu, B., & Galitsky, B. (2022). Graph-Based Skill Matching for Job Recommendation. IEEE Transactions on Knowledge and Data Engineering.
8. Malinowski, J., Kim, T., Wendt, O., & Weitzel, T. (2006). Matching People and Jobs: A Bilateral Recommendation Approach. HICSS.
9. European Commission. (2020). ESCO — European Skills, Competences, Qualifications and Occupations. Publications Office of the EU.
10. Lehmann, J. et al. (2015). DBpedia — A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web Journal, 6(2), 167–195.
11. Xu, J., Zhu, X., & Bellmore, A. (2022). Implicit Skill Inference from Job Descriptions Using Word Embed-dings. Expert Systems with Applications, 191, 116265.
12. Handshake. (2023). Campus Recruiting Platform White Paper. Handshake Inc.
13. Lindbergh and H. K. H. Lee, “Optimization under constraints by applying an asymmetric entropy measure,” J. Comput. Graph. Statistic., vol. 24, no. 2, pp. 379–393, Jun. 2015, doi: 10.1080/10618600.2014.901225.
14. Rieder, Engines of Order: A Mechanology of Algorithmic Techniques. Amsterdam, Netherlands: Amsterdam Univ. Press, 2020.
15. Boglaev, “A numerical method for solving nonlinear integro-differential equations of Fredholm type,” J. Comput. Math., vol. 34, no. 3, pp. 262–284, May 2016, doi: 10.4208/jcm.1512-m2015-0241.