IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

AI-Powered Resume Analysis and Job Matching System

Arun kumar P, Kaviyarasan V, Mohamed Nabil S
UG Students, Department of AI & Data Science
Sengunthar Engineering College (Autonomous), Tiruchengode, India
arunk235696@gmail.com , Kavikavi7143@gmail.com
mohamednabil292638@gmail.com

N.Indhuja
Assistant Professor, Department of AI & Data Science
Sengunthar Engineering College (Autonomous), Tiruchengode, India
nindhuja.aids@scteng.co.in
https://orcid.org/0009-0002-9513-2438

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10091
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10091
Received: 30, January 2026, Revised: 13, February 2026, Accepted: 28 February 2026 Published Online: 25 March 2026
https://www.irjcs.com/volumes/Vol13/iss-03/12.CSMR26.MRCS10091.pdf
Article Citation: Arun,Kaviyarasan,Mohamed,Indhuja(2026),AI-Powered Resume Analysis and Job Matching System,IRJCS :International Research Journal of Computer Science, Volume 13,Issue 03 of 2026 pages 161-168
Doi:> https://doi.org/10.26562/irjcs.2026.v1303.12 **BibTeX Key Arun@2026AI-Powered**
Orcid: https://orcid.org/0009-0004-9398-7488
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1303.12 The journal's official abbreviation is IRJCS.

Abstract: The growing volume of digital job applications has overwhelmed traditional manual resume screening processes, creating an urgent need for intelligent, scalable automation in modern recruitment. This paper presents an AI-Powered Resume Analysis and Intelligent Job Matching System (RAJMS) that integrates Natural Language Processing, TF-IDF vectorization, cosine similarity matching, and machine learning to automate end-to-end candidate evaluation. The system accepts candidate resumes in PDF, DOCX, and plain-text formats alongside free-form job description inputs, applies named entity recognition to extract structured skill, education, and experience entities, and transforms the extracted content into high-dimensional TF-IDF feature vectors. Cosine similarity between candidate vectors and job-description vectors produces quantitative matching scores enabling objective, bias-free candidate ranking. An optional BM25 probabilistic ranking ensemble supplements TF-IDF matching for improved precision on domain-specific vocabulary. Evaluated on a dataset of 3,200 resumes across ten industry domains, the proposed RAJMS achieves candidate matching accuracy of 91.6%, outperforming conventional keyword- based Applicant Tracking Systems by 22.3 percentage points, and delivers end-to-end short listing for 500 resumes in under 3 seconds, demonstrating the practical viability of NLP-driven recruitment automation.
Index Terms: Resume Analysis, Natural Language Processing, Job Matching, Machine Learning, Candidate Recommendation System, TF-IDF, Cosine Similarity, Applicant Tracking, Skill Extraction, Automated Recruitment.

I. INTRODUCTION

A. The Recruitment Automation Challenge: The digital transformation of the labour market has produced an unprecedented surge in online job applications. Major portals such as LinkedIn, Naukri, and Indeed collectively host billions of applications annually, with individual corporate openings routinely attracting thousands of resume submissions [1,2].This volume fundamentally overwhelms traditional recruitment work flows, in which human resource professionals manually review each submitted resume to assess candidate qualifications against role requirements. Industry surveys estimate that a typical recruiter spends six to seven seconds on initial resume screening, making rapid, consistent, and fair assessment practically impossible at scale. The operational cost of manual screening is substantial, with average time-to-hire estimated at 36days and cost-per-hire exceeding $4,000 for many organizations [3].

B. Limitations of Existing Systems

Manual resume screening introduces three systematic limitations. First, subjectivity and inter-rater inconsistency: different recruiters apply different evaluation criteria, producing variable short listing decisions that erode hiring quality. Second, unconscious bias: research has documented that resume attributes irrelevant to job performance including candidate name and educational institution prestige influence recruiter judgments, reducing diversity in shortlisted pools [1].Third, format variability: resumes are submitted in dozens of formats ranging from structured templates to free-form narrative documents, making consistent information extraction challenging for manual reviewers and simple keyword-matching systems alike [4,3].

C. Role of NLP and Machine Learning

Natural Language Processing and machine learning provide the computational foundation for intelligent resume analysis.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

NLP techniques including named entity recognition, part-of- speech tagging, and TF-IDF vectorization enable machines to extract semantically meaningful information from unstructured resume text [5,6]. Machine learning similarity metrics quantify the alignment between candidate profiles and job requirements, enabling objective ranking that scales to any number of candidates without degradation in consistency. These capabilities address all three limitations of manual screening: automation removes subjectivity, algorithmic scoring removes demographic bias pathways, and multi-format parsing removes format barriers.

D. Paper Contributions

This paper makes four contributions:(i) the RAJMS frame- work integrating multi-format resume parsing, spaCy-based NLP extraction, TF-IDF vectorization, and cosine similarity matching in an end-to-end pipeline; (ii)aBM25 ensembles coring mechanism that improves precision on domain-specific vocabulary; (iii) a Stream lit-based recruiter dashboard with skill- gap analysis and downloadable shortlist reports; and (iv) comprehensive evaluation on 3,200 resumes across ten industry domains confirming state-of-the-art matching accuracy.

II. LITERATUREREVIEW

A. Resume Parsing and Information Extraction

Early automated resume parsing systems applied rule-based information extraction using regular expressions and section-header keyword matching to segment resumes into structured fields: name, contact information, education, work experience, and skills. While effective for highly structured resume tem- plates, these approaches are brittle under the format diversity of real-world submissions, requiring continuous manual maintenance as resume conventions evolve[3,11]. Kovaˇcˇ and Podjed[4] demonstrated that NLP pipelines in corporating tokenization, lemmatization, and TF-IDF vectorization significantly outperform rule-based approaches in recall of relevant resume content, particularly for resumes describing skills in non-standard or domain-specific terminology.

B. NLP-Based Recruitment Tools

Jiechieu and Tsopze [5] introduced a multi-label CNN architecture for simultaneous skill extraction and job recommendation from resumes, achieving an F1-score of 0.87 across 50 skill categories on the Kaggle resume dataset. Harsha and Manaswi [6] evaluated cosine similarity as the primary matching metric for TF-IDF-vectorized resume-job pairs, confirming its effectiveness across ten industry categories. Daveetal.[7]combined TF-IDF with Latent Semantic Analysis for resume-job matching, observing that semantic dimensionality reduction via SVD improves precision when resumes and job descriptions use different vocabulary for identical competencies, with average precision at rank10 improving by11.4 % over pure TF-IDF cosine similarity.

C. Job Recommendation and ML Hiring Systems

Sinha et al. [8] demonstrated that BERT-based sentence embeddings improve resume-job matching precision particularly for domain-specific technical roles where skill synonyms are critical, outperforming TF-IDF cosine similarity by 8.7 percentage points in precision@10on software engineering job descriptions. Li et al. [10] surveyed job recommendation approaches, identifying content-based NLP matching as the best-performing paradigm for new-candidate cold-start scenarios. Qinetal.[12] proposed an ability-aware neural network for person-job fit, achieving state-of-the-art performance on the BOSS Zhipin recruitment dataset. Raghavanetal.[1] provided a critical algorithmic audit demonstrating that ML-based resume screening can amplify demographic biases unless explicit fairness constraints are incorporated, motivating the bias-reduction design principles in the proposed RAJMS system [13, 9].

III. PROBLEMSTATEMENT

A. Formal Problem Definition: The resume-job matching problem is formalized as follows. Let a corpus of N candidate resumes be denoted $R = \{r1,r2,...,rN\}$ And let J denote a target job description. Each resume r_i and job description J are unstructured text documents. The objective is to compute a relevance score s_i = match (r_i,J) for each candidate and return an ordered ranking σ such that $s_{\sigma(1)} \geq s_{\sigma(2)} \geq \cdots \geq s_{\sigma(N)}$, where higher-ranked candidates are more qualified for the position [4].The system must process each resume within a sub-3-second total latency for batches of up to 500 resumes to support interactive recruiter use.

B. Limitations of Current Systems

Three deficiencies characterize existing hiring systems. First, keyword-based ATS platforms such as Taleo and Work- day filter resumes using Boolean keyword queries that lack semantic understanding, causing qualified candidates using domain-equivalent non-standard terminology to be incorrectly eliminated a problem that disproportionately affects non-native English speakers and career changers [1]. Second, format variability: resumes submitted in diverse formats require robust multi-format parsing before text analysis can begin. Third, cold- start limitation: job portal recommendation engines based on collaborative filtering fail for new candidates with limited application history, creating disadvantage for early-career applicants who may be the most suitable candidates for entry-level roles [10].

IV. PROPOSED SYSTEM

A. System Overview

The AI-Powered Resume Analysis and Intelligent Job Matching System (RAJMS) is a fully automated end-to-end candidate evaluation platform that processes resumes from upload through NLP parsing, feature vectorization, similarity scoring, and ranked recommendation output. The system operates without any manual recruiter intervention in the short listing work- flow, compressing time-to-shortlist from days to seconds.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

B. Resume Upload and Multi-Format Parsing

The Resume Upload Module accepts candidate resumes in PDF, DOCX, and plain-text formats. PDF extraction uses PyMu PDF's high-fidelity text extraction engine which pre- serves layout structure.DOCX extraction uses python-docx to parse XML content from Office Open XML archives. Scanned PDF documents identified by low text yield are routed to Tesser-act5.0 OCR preprocessing before NLP processing. A multi-file batch upload interface allows recruiters to submit hundreds of resumes simultaneously against a single job description, with a progress indicator tracking per-file status.

C. NLP-Based Resume Parsing and Entity Extraction

The NLP parsing module applies a five-stage processing pipeline to each extracted resume text. Named Entity Recognition using spaCy's encore web_sm model identifies PERSON, ORG, DATE, and GPE entities. A domain-specific skill extraction module maps skill indicating tokens to acurated ontology of 2,400 technical and soft skills across twelve industry categories through exact-match and fuzzy-match lookup using RapidFuzz. Education entities (degree level, institution, graduation year) and work experience (employer names, job titles, employment duration) are extracted via combined NER and regex pattern matching.

D. Candidate Ranking and Recommendation

The job matching engine computes TF-IDF feature vectors for all resume documents and the job description jointly. Cosine similarity between each resume vector and the job-description vector produces a scalar matching scores $i \in [0,1]$.

An ensemble mode combines TF-IDF cosine similarity with BM25 probabilistic rankings $s = \alpha \cdot s + (1-\alpha)s$, with

E. Skill and Feature Extraction

The skill extraction module maps resume tokens to acurated ontology of 2,400 skills across twelve industry categories using exact-match and fuzzy-match look up. Education entities includi iiα=0.6 determined through cross-validation. Candidates are sorted by descending score; the Stream l it dashboard displays the top-k candidates with match scores, skill-match break downs, and downloadable shortlist reports.

V. SYSTEM ARCHITECTURE

The RAJMS system pipeline is organized into four functional layers: an Input Layer accepting multi-format resumes and job descriptions; a Processing Layer applying NLP text extraction and structured entity parsing; a Matching Layer performing TF-IDF vectorization and cosine similarity scoring; and an Output Layer delivering ranked candidate dashboards and export reports. The Input Layer accepts resumes in PDF, DOCX, and plain- text formats alongside job description text. A batch upload interface processes up to 500 resumes per job description submission. The Processing Layer applies multi-format text extraction, NLP preprocessing, and skill entity extraction to produce structured candidate feature profiles from raw resume text. The Matching Layer performs joint TF-IDF vectorization of all resumes and the job description, followed by pair wise cosine similarity computation and optional BM25 ensemble scoringto produce quantitative match scores. The Output Layer renders ranked candidate tables on the Streamlit dashboard with score breakdowns; CSV and PDF export functions provide arte facts for recruiter workflow integration and audit trail maintenance. The system applies four engineering principles: latency-first design with sub-3-second processing for 500 resumes; format- agnostic multi-parser architecture; modular independence enabling component-level upgrades; and full audit traceability with per-candidate match score logs and model version identifiers for regulatory compliance. Degree level, institution name, and graduation year are extracted via combined NER and regex patterns. Work experience entities including employer names, job titles, and employment duration in months are extracted similarly. Boolean and count- based features are generated per skill category (programming languages, frameworks, tools, soft skills, domain knowledge). Together, these structured features augment the TF-IDF bag-of- words representation to form the final feature vector used for similarity computation.

TF-IDF Vectorization

After preprocessing, each resume and the job description are represented as TF-IDF weighted bag-of-words vectors. The TF- IDF weight for term t in document d within corpus D is computed as:

$$\text{Tfidf}(t,d,D) = \text{tf}(t,d) \times \log \frac{|D|+1}{|\{d' : t \in d'\}|+1} + 1 \qquad (1)$$

where tf(t,d) is the normalized term frequency with sub linear scaling and the denominator applies +1 smoothing to prevent division by zero. Scikit-learn's Tfidf Vectorizer with unigram and bigram token range, L2 normalization, and sublinear TF scaling is fitted on the full resume corpus and applied uniformly to job descriptions.

VI. MACHINE LEARNING MODEL

A.TF-IDF Cosine Similarity Matching

The fitted TF-IDF vocabulary on the combined corpus contains up to 15,000 terms capturing the most discriminative unigrams and bigrams. Each document is represented as an L2- normalized vector in this high-dimensional feature space. Since both vectors are L2-normalized, cosine similarity reduces to a dot product enabling efficient batch computation:

I. METHODOLOGY

A. Text Preprocessing Pipeline

$$s(r,j) = \frac{\mathbf{v}_{r_i} \cdot \mathbf{v}_J}{\|\mathbf{v}_{r_i}\| \cdot \|\mathbf{v}_J\|} = \mathbf{v}_{r_i} \cdot \mathbf{v}_J \qquad (2)$$

The text preprocessing pipeline transforms raw resume text through six sequential normalization stages. Lower casing converts all text to lowercase for case-insensitive token matching. Punctuation removal eliminates non-alphanumeric characters using regex substitution, preserving hyphens in compound technical terms.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

Tokenization applies NLTK's word_tokenize to split text into individual tokens, correctly handling contractions The full score matrix$\mathbf{s}=\mathbf{V}_R \cdot \mathbf{v}_J$ is computed via NumPy matrix multiplication, enabling sub-second scoring of thousands of resumes in a single vectorized operation [13].

B. BM25 Probabilistic Ranking

The optional BM25ensemblesupplements cosine similarity with probabilistic term weighting. For term t in document d with average document length |d|: and abbreviations. Stop-word removal applies NLTK's179-word English stop-word corpus, eliminating high-frequency function words that contribute nod is criminative information to TF-IDF representations. Lemmatization uses spaCy's morpho logical analyser to reduce inflected forms to their base lemma (e.g.,"managing"→"manage","databases"→"database"),ensuring that conjugated and plural forms are treated as identical tokens. Named Entity Recognition then tags PERSON, ORG, DATE, SKILL, and QUALIFICATION entities for dow stream structured extraction. With saturation parameter k_1=1.5 and length normalization b= 0.75.BM25 handles varying document lengths and rareterm Saturation more effectively than TF-IDF, complementing cosine similarity in the ensemble. The ensemble weight α=0.6 was determined through 5-foldcross-validation on the validation set, maximizing NDCG@10.

$$\text{BM25}(t,d) = \frac{\text{tf}(t,d)(k_1+1)}{\text{tf}(t,d)+k_1\left(1-b+b\frac{|d|}{|d|}\right)} \cdot \log\frac{|D|-\text{df}(t)+0.5}{\text{df}(t)+0.5} \quad (3)$$

C. Candidate Ranking Algorithm

The ranking function returns the ordered candidate list $\sigma = \text{argsort}(-\mathbf{s})$.The top-k ranked candidates (default k=10, configurable) are presented with their match scores, skill overlap break downs showing intersection of extracted resume skills and job-description requirements, and missing skill gap summaries to inform interview question targeting. The system also computes a domain classification score using a trained Naïve Bayes classifier on the TF-IDF vectors, enabling cross-domain applicant filtering when recruiters specify vertical restrictions.

D. Model Training and Validation

The TF-IDF vocabulary is fitted on the complete resume corpus using Scikit-learn's Tfidf Vectorizer with max_features =15000,ngram_range=(1,2), sublinear_tf=True,andmin_df=2.No supervised training is required for the base cosine similarity pipeline. The BM25 ensemble weight α is optimized via 5-fold cross-validation using NDCG@10 as the objective, running 50 trials over $\alpha \in [0.4,0.8]$. The resulting fitted vocabulary covers 94.7% of Term occurrences in the evaluation corpusata density of 0.8%, enabling efficient parse matrix storage and fast dot-product similarity computation.

VII. RESUME–JOB MATCHING ALGORITHM

A. Matching Pipeline Overview

The RAJMS matching algorithm executes a five-step sequential pipeline for each batch submission. Step 1: multi-format text extraction from all uploaded resumes (PyMuPDF, pythondocx, or plain text reader as appropriate).Step 2: NLP preprocessing of extracted text through the six-stage pipeline described in Section VI- A.Step3: structured entity extraction to build per-candidate skill, education, and experience feature profiles. Step 4: joint TF-IDF vectorization of all resume and job-description text using the fitted vocabulary. Step 5: cosine similarity computation and BM25 ensemble scoring to produce the final ranked candidate list. The complete pipeline executes in under 2.8 seconds for 500 resumes on reference hardware.

B. Skill Ontology Matching

Beyond TF-IDF text similarity, the system performs exact and fuzzy skill ontology matching to capture semantic Equivalences not captured by lexical overlap. Extracted resume skill tokens are matched against the 2,400-skill ontology using Rapid- Fuzz with an 85% similarity threshold, handling minor spelling variations and abbreviations. The resulting skill-match coverage score the fraction of job-required skills present in the candidate's profile is incorporated as a weighted additive bonus:

Table1 – Resume Feature Extraction Fields

Field	Method	Type	Example
Candidate Name	NER (PERSON)	String Arunkumar P	Email / Phone RegexString arunk@gmail.com
Education	DegreeNER + Ontology Ordinal	B.Tech, M.Tech	Institution NER String
(ORG)	Anna University	Graduation Year NER (DATE)	Integer2024
Job Titles	NER + Keyword List	Data Analyst Employers	NER (ORG) List Infosys, TCS
Experience (yrs)	DATE arithmetic Float	3.5 years Technical Skills	Skill Ontology List Python, SQL
Soft Skills	Skill Ontology List+ Leadership	Certifications Keyword +ERList	AWS Certified
Languages	Keyword	MatchList	English,Tamil

VIII. DATASET AND FEATURES

A. Dataset Composition

The RAJMS system is trained and evaluated on a curated corpus of 3,200 resumes spanning ten industry domains collected from anonymized public resume datasets.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

Each resume is paired with a set of job description queries and annotated with binary relevance labels by three domain-expert annotators (Cohen's κ = 0.81).Table 1 presents the complete feature extraction fields used for candidate profiling.

B. Feature Engineering and TF-IDF Vocabulary

The fitted TF-IDF vocabulary on the 3,200-resume corpus contains 47,312 unique unigrams and 89,641 unique bigrams before truncation. With max_features=15000, the vocabulary retainsthe15,000highestdocument-frequencyterms,covering 94.7% of term occurrences in the corpus. The resulting feature matrix is highly sparse at 0.8% density, enabling efficient Compressed Sparse Row storage and fast dot-product similarity computation.

C. Domain and Vocabulary Analysis

Domain-level vocabulary analysis reveals that technical do- mains such as software engineering and data science achieve higher vocabulary stability across the corpus thetop-500 high- IDF terms explain 62% of discriminative information in these domains while soft skill and general management domain sex hibit greater lexical variability requiring wider vocabulary coverage. Bigram features contribute disproportionate discriminative value for skill identification: phrases such as "machine learning", "data analysis", "project management", and "cloud infrastructure" cannot be captured by unigrams alone, motivate $_{S}final_{=S}ens_{+\beta}.|Sr_{i}\cap S_{J}|(4)$

where $Sr_{i}$is the set of skills extracted from resume r_{i}, $S_{J}$is the set of skills required by job description J, and β =0.15 is the skill coverage bonus weight tuned on the validation set. Class distribution analysis of the 3,200-resume corpus shows that software engineering (20%), data science (15%), and marketing (10%) are the three most common domains, while le-gal (3%) and operations (4%) are the smallest, requiring the domain-stratified evaluation protocol to prevent performance inflation from majority-domain dominance.

IX.SYSTEM IMPLEMENTATION

A. Technology Stack

The RAJMS system is implemented in Python3.10. spaCy 3.5 with the encore web sm model and NLTK 3.8 provide NLP processing. Scikit-learn 1.3 handles TF-IDF vectorization and cosine similarity computation. The rank_bm25 library 0.2.2 implements BM25 scoring. PyMuPDF 1.23 extracts text from PDFs; python-docx1.0 handles DOCX parsing. The web interface is implemented in Stream lit1.28. Pandas2.1 and NumPy 1.26 handle tabular data management and numerical computation. Report Lab4.0 generates PDF shortlist reports. The system is containerized using Docker 24 for reproducible cross-platform deployment on Windows, Linux, and macOS.

B. Implementation Workflow

The complete processing workflow executes six sequential steps. First, the recruiter uploads a batch of resumes and enters the job description through the Stream lit interface. Second, the multi-format parser extracts plain text from each resume file. Third, the NLP pipeline tokenizes, normalizes, and extracts named entities. Fourth, the TF-IDF vectorizer transforms all text to feature vectors. Fifth, cosine similarity and BM25 ensembles coring ranks all candidates. Sixth, the ranked result stable, skill-gap analysis, and downloadable CSV and PDF reports are rendered on the dashboard. The full pipeline completes in under 2.8 seconds for 500 resumes on a standard workstation meeting the hardware requirements of Intel Core i5 or above,8 GB RAM, and 500 GB storage.

C. Database and Audit Trail

A SQLite database stores processed resume records including extracted entities, serialized TF-IDF vectors as NumPy ar-rays, job description records, and historical match results with timestamps. The schema supports incremental resume corpus updates: new resumes are processed and appended without requiring full vocabulary refitting, using the frozen vocabulary from the initial training corpus. Sessionlogsareretainedfor90 days to support audit trail requirements and equal employment opportunity reporting .A Grafana monitoring dashboard tracks system throughput, processing latency per resume, vocabulary hit rate, and error rates with 5-minute update intervals.

D. Hardware Requirements and Deployment

The RAJMS system requires an Intel Core i5 or AMD Ryzen 5 processor, minimum 8GBRAM (16GB recommended for batch sizes exceeding 1,000 resumes), and 500 GB storage. No GPU is required for the core TF-IDF cosine similarity pipeline; the system is entirely CPU-bound and scales linearly with resume corpus size. The Docker container image is approximately 2.1GB including all Python dependencies. Deployment on cloud platforms such as AWS EC2 or Google Cloud Run is supported via the standard Docker image, enabling elastic scaling for high-volume recruitment events. End-to-end processing of 500 resumes against a single job description completes in under 2.8 seconds on the reference hardware configuration, meeting the interactive use latency requirement for recruiter-facing short listing workflows.

Table 2 – Model Performance Metrics

Method	Acc.(%)	P@10	R@10	F1
Keyword ATS	69.3	0.651	0.714	0.681
TF-IDF + Cosine	85.1	0.843	0.861	0.852
BM25 Only	86.4	0.857	0.872	0.864
TF-IDF+BM25Ens.	89.7	0.891	0.903	0.897
RAJMS (Ours)	91.6	0.912	0.921	0.916

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

Table 3- Candidate Matching Accuracy

Metric	RAJMS	Keyword ATS
Matching Accuracy	91.6%	69.3%
Precision@10	0.912	0.651
Recall@10	0.921	0.714
F1-Score	0.916	0.681
NDCG@10	0.904	0.623
Mean Reciprocal Rank	0.881	0.594
Processing Time (500)	2.8 s	0.4s

X. EXPERIMENTAL RESULTS

A. Evaluation Protocol

The system is evaluated on the 3,200-resume corpus across ten industry domains. Four baselines are compared: (1) Key-word ATS using Boolean job-description keyword matching;
(2) TF-IDF cosine similarity alone; (3) BM25 ranking alone; and (4) TF-IDF plus BM25 ensemble without skill ontology bonus. Table 2 presents comparative performance on matching accuracy, precision@10, recall@10, and F1-score.

B. Matching Accuracy and Ranking Quality

The RAJMS system achieves 91.6% matching accuracy, a 22.3 percentage point improvement over keyword ATS. The inclusion of the skill ontology bonus in the final RAJMS score adds1.9percentage points over the TF-IDF plus BM25 ensemble alone, confirming that structured skill matching provides incremental value beyond lexical similarity. Table 3 presents additional ranking quality and operational metrics. NDCG@10 of 0.904 confirms that the highest-quality candidates are consistently ranked at the top of the shortlist, the most operationally relevant property for recruiter-facing systems. Mean Reciprocal Rank of 0.881 indicates that the best- matched candidate appears as the first result in the majority of evaluated queries. The 2.8-second processing time represents a dramatic compression of recruiter time-to-shortlist from the industry average of several days.

C. Domain-Specific Analysis

Performance varies modestly across industry domains. Software engineering achieves the highest NDCG@10of0.941, reflecting the well-standardized technical vocabulary in that do- main. Healthcare achieves the lowestat 0.871,reflecting greater lexical diversity in clinical skill descriptions. Data science and finance achieve intermediate NDCG@10 scores of 0.928 and 0.912 respectively. Across all ten domains, the RAJMS system out performs keyword ATS by a minimum of17.8 percentage points, confirming broad generalizability of the NLP-based matching approach.

D. Ablation Study

An ablation study quantifies the contribution of individual system components. Removing the BM25 ensemble and using TF-IDF cosine similarity alone reduces matching accuracy by 6.5 percentage points (91.6% to 85.1%), confirming that BM25 probabilistic weighting adds substantial value for domain-specific vocabulary. Removing the skill ontology bonus reduces accuracy by a further 1.9 points, showing that structured skill matching provides incremental value beyond lexical TF-IDF overlap. Removing lemmatization from the preprocessing pipeline degrades NDCG@10 by 3.2 points, validating the importance of morphological normalization for consistent to- ken matching. Replacing spaCy NER with regex only extraction reduces skill extraction recall by11.4%, confirming the importance of learned entity recognition for complex multi-entity resume sentences. These results confirm that each component contributes independently to the final system performance, and that the integrated RAJMS pipeline provides compounding improvements over any single-component baseline.

XI. ADVANTAGES

The RAJMS framework delivers six measurable advantages over conventional recruitment systems, each directly addressing a documented limitation of existing approaches.

1) Faster Recruitment Process: End-to-end short listing of 500 candidates in 2.8 seconds versus the industry average of several days dramatically compresses time-to-hire and enables recruiters to engage candidates while their application interest remains high. This speed advantage is especially critical for high-volume hiring programs in retail, logistics, and financial services where seasonal demand creates time-sensitive staffing requirements.
2) Automated Multi-Format Resume Filtering: The system handles PDF,DOCX, and plain-text resumes without requiring candidates to conform to any particular template, eliminating the format-compatibility bottleneck that causes keyword-based ATS platforms to miss qualified candidates who submit non- standard resumes.
3) Semantic Candidate-Job Matching: TF-IDF matching with BM25 ensemble scoring and skill ontology lookup identifies qualified candidates describing skills in non-standard or domain-specific terminology, improving recall of genuinely qualified candidates by 22.3 percentage points over keyword- matching baselines.
4) Reduced Recruitment Bias: Scoring candidates on extracted textual content without reference to demographic identifiers including name, gender, or educational institution removes the primary pathway for unconscious bias, supporting fairer and more diverse hiring outcomes [1].
5) Transparent and Auditable Scoring: The match score and skill-level breakdown provided for each ranked candidate ex- plain the ranking in human-interpretable terms, fulfilling audit trail requirements for regulated industries and equal employment opportunity compliance reporting.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

6) Scalable Modular Architecture: Docker containerization and sparse matrix computation support arbitrarily large candidate pools with linear computational cost. The modular design allows individual components LP parser, vectorizer, ranker, dashboard to be upgraded independently.

XII. FUTUREWORK

A. DeepLearning Resume Parsing

Future work will replace the current spaCy NER and regexbased extraction with a fine-tuned BERT-based token classifica tion model trained on a labeled corpus of resume text annotated with skill, education, and experience entities[14]. BERT'sbi directional contextual embeddings will improve entity boundary detection in complex multi-entity sentences and resolve abbreviation ambiguities that confound pattern-based extraction.

B. Semantic Skill Matching and Knowledge Graphs

Current skill matching relies on fuzzy lexical overlap against a static ontology. Future work will introduce semantic matching using Sentence-BERT embeddings that capture contextual proximity between skill descriptions, enabling matches between semantically equivalent but lexically distinct skill phrases. Integration with structured skill ontologies such as ESCO and O*NET will provide hierarchical skill relationships for improved cross-domain generalization [14].

C. Online Job Portal Integration and Mobile Application

Integration with major job portal APIs—LinkedIn, Indeed, and Naukri via OAuth 2.0 authenticated REST connections will enable real-time bidirectional data flow, automatically ingesting newly posted job descriptions and submitted resumes into the matching pipeline. A mobile companion application for iOS and Android will allow job seekers to receive instant match score feedback and personalized skill-gap recommendations [12, 10].

XIII. CONCLUSION

This paper presented the AI-Powered Resume Analysis and Intelligent Job Matching System (RAJMS), a comprehensive NLP and machine learning platform automating the complete recruitment short listing workflow from multi-format resume upload through structured entity extraction, TF-IDF vectorization, cosine similarity and BM25 ensemble scoring, skill ontology matching, and ranked candidate recommendation. Evaluated on 3,200resumesacrosstenindustry domains, RAJMS achieves a matching accuracy of 91.6%—a22.3 percentage point improvement over keyword-based ATS with NDCG@10 of 0.904 and Mean Reciprocal Rank of 0.881 confirming consistent top ranked quality. End-to-end processing of 500 resumes completes in 2.8 seconds. The system's six-stage NLP preprocessing pipeline, 2,400- skill ontology, transparent match score with skill-level break- downs, and format-agnostic multi-parser architecture collectively deliver measurable improvements in recruitment speed, accuracy, fairness, and interpretability. Component-level ablation analysis confirms that each system element contributes in- dependently: the BM25 ensemble adds 6.5 percentage points over TF-IDF cosine alone, skill ontology matching adds a further1.9points,andlemmatizationimprovesNDCG@10by3.2 points. The RAJMS framework provides a practical, deployable, and scalable solution for corporate HR platforms, online job portals, campus recruitment, and public employment services. By automating the end-to-end short listing work flow from resume upload to ranked candidate output in under3 seconds the system substantially reduces recruiter workload while improving the quality, consistency, and fairness of hiring decisions across all evaluated industry domains. Future extensions incorporating BERT-based contextual semantic parsing, federated learning for privacy-preserving cross-organizational model improvement, live job portal API integration, and mobile candidate feed- back applications will further expand the system's capability, reach, and impact across the modern digital recruitment ecosystem.

REFERENCES

1. M.Raghavan,S.Barocas,J.Kleinberg,andK.Levy,"Mitigating bias in algorithmic hiring: Evaluating claims and practices," in Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, Barcelona,Spain, 2020, pp. 469–481.
2. I.Naim, M.I.Tanveer, D.Gildea, and M.E.Hoque, "Automated analysis and prediction of job interview performance," IEEE Transactions on Affective Computing, vol. 9, no. 2, pp. 191–204, 2018.
3. X.Chen, M.Xu, and C.Ding, "RESUME: Information extraction from resumes," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 2018, pp. 1–10.
4. M.Kovaˇcˇ and D.Podjed, "Natural language processing for human resource management: Resume screening and job matching," Journal of Information and Organizational Sciences, vol. 44, no. 2, pp. 223–245, 2020.
5. K.F.F.Jiechieu and N.Tsopze, "Skills prediction based on multi-label resume classification using CNN with model predictions explanation," Neural Computing and Applications, vol. 33, pp. 6323–6337, 2021.
6. B.Harsha and S.Manaswi,"Resume analysis and job recommendation using cosine similarity and machine learning," in Proceedings of the IEEE International Conference on Intelligent Computing and Control Systems, Madurai, India, 2021, pp. 1–7.
7. V.Dave,S.Gaur,andP.Dhande,"Combined approach for resume ranking using TF-IDF and latent semantic analysis," in Proceedings of the IEEE International Conference on Inventive Research in Computing Applications, Coimbatore, India, 2018, pp. 1–6.
8. P.Sinha,S.Yadav,andA.Singh,"NLP-based resume screening and jobmatching using BERT embeddings," in Proceedings of the IEEE International Conference on Computing, Communication and Automation, Greater Noida, India,2021,pp.1–6.
9. J.Zhang,C.Tan,andX.Zhang,"Resume information extraction with the cascaded hybrid model," in Proceedings of the Annual Meeting of the Association for Computational Linguistics, Seattle, WA, 2020, pp. 6429–6439.
10. X.Li, L.Lyu, and Q.Liu, "A survey of job recommendation in recruitment system," Journal of Physics: Conference Series, vol. 1213, no. 4, p. 042019, 2019.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.12

11. T.Schmitt and A.Ramsay, "Automated resume parsing and candidate ranking: A hybrid NLP approach," in Proceedings of the International Conference on Language Resources and Evaluation, Istanbul, Turkey, 2012, pp. 1–7.
12. C.Qin,H.Zhu,T.Xu,C.Zhu,L.Jiang,E.Chen,and H.Xiong,"Enhancing person-job fitfor talent recruitment: An ability-aware neural network approach," in Proceedngs of the ACM SIGIR Conference on Research and
13. Development in Information Retrieval, Ann Arbor,MI,2018, pp.25–34.
14. C.D.Manning, P.Raghavan, and H.Schütze, Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK, 2008.
15. J.Devlin, M.W. Chang, K.Lee, and K.Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of NAACL Human Language Technologies, Minneapolis, MN,2019, pp.4171–4186.
16. D.Luo,J.Yang,M.Liu,andY.Liu,"Resume GAN: An optimized deep representation learning frame work for talent-job fit via adversarial learning," in Proceedings of the ACM International Conference on Information and Knowledge Management, Beijing, China,2019,pp.1–10.