



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: <https://doi.org/10.22214/ijraset.2026.85033>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

An Explainable Rule-Based Resume-Job Description Skill Matching System with LLM-Assisted Feedback

S. Jathin Chowdary¹, K. Ravi Kumar², S. Mallikarjun³, N. Srishma⁴, R. Vivek⁵

^{1,3,4,5} Undergraduate Students, Hyderabad Institute of Technology and Management, Hyderabad, Telangana, India

² Associate Professor, Department of CSE, Hyderabad Institute of Technology and Management, Hyderabad, Telangana, India

Abstract: Automated resume screening and Applicant Tracking Systems (ATS) increasingly incorporate machine learning components and neural language models. While deep learning methods can capture complex linguistic patterns, their computational overhead, potential opacity, and sensitivity to unvetted training distributions pose practical and ethical challenges for lightweight, transparent academic deployments. In this paper, we describe an open, lightweight, and explainable resume-job description matching system that couples deterministic rule-based information extraction with optional Large Language Model (LLM) qualitative feedback. The core pipeline processes PDF, DOCX, and plain-text resumes, detects functional sections, extracts technical competencies using an auditable 108-skill taxonomy, and evaluates resume-job description compatibility using explicit set operations and a bounded 100-point heuristic score covering Skills, Experience, Projects, Education, and Certifications. To benchmark the pipeline without making real-world recruitment claims, we developed a deterministic synthetic evaluation harness containing 100 resumes, 20 job descriptions, 320 evaluation pairs, and 20 candidate ranking cohorts. Under the synthetic benchmark, the system achieved F1-scores of 0.791 for skill extraction, 0.827 for education, 0.772 for experience, 0.759 for projects, and 0.865 for missing-skill identification. Robustness testing exposed 0.0% recall when experience titles and dates were separated across lines and complete section collapse under unsupported headings. Synonym normalization increased skill recall from 0.896 to 0.968 under controlled conditions. The results characterize the behavior and limitations of the proposed deterministic baseline and provide a reproducible basis for future evaluation with real, anonymized and human-annotated resume data. The benchmark also reports multi-format consistency and controlled baseline comparisons to expose formatting sensitivity and vocabulary-related error patterns.

Keywords: Resume Parsing, Information Extraction, Skill Matching, Applicant Tracking Systems, Explainable AI, Synthetic Benchmarking.

I. INTRODUCTION

The recruitment landscape processes large volumes of curriculum vitae (CVs) and job descriptions across digital employment platforms. To handle high application volumes, organizations frequently rely on Applicant Tracking Systems (ATS) to assist in parsing, filtering, and organizing applicant pools. Over the past decade, some recruitment technology has shifted from basic keyword search toward complex machine learning, natural language processing (NLP), and deep representation learning pipelines, including convolutional networks, recurrent architectures, and transformer-based sentence encoders [5], [10].

However, prior studies have reported that complex or opaque predictive screening pipelines can introduce practical and ethical challenges:

- 1) Decision Opacity: Some deep neural architectures operate with limited interpretability, which can make it difficult for candidates or auditing bodies to discern why a specific resume received a particular score [6].
- 2) Computational and Governance Overheads: Training and fine-tuning domain-specific language models can require substantial compute resources and proprietary datasets of historical hiring records, which are rarely accessible to independent academic researchers due to privacy and governance constraints.
- 3) Concerns Regarding Predictive Validity: Prior empirical audits of automated screening tools have raised concerns regarding the degree to which algorithmic scores correlate with validated job performance, noting that proxy variables may replicate historical hiring patterns [6].

These considerations motivate the investigation of **explainable, deterministic, and lightweight screening architectures** where extracted entities, matching decisions, and score components can be directly inspected, audited, and verified by human operators. In this work, we present an explainable resume–job description matching system that strictly decouples deterministic rule-based matching from optional LLM-assisted qualitative feedback. The deterministic pipeline performs multi-format document ingestion, structural section detection, regular-expression-based education and experience extraction, and dictionary-driven skill identification. Matches and gaps are computed via transparent set intersections, producing an auditable heuristic ATS-style score bounded between 0 and 100. Optional generative capabilities (e.g., bullet enhancement and tailored cover letter generation via Google Gemini) are isolated as non-critical advisory services; if the external API is unconfigured or unreachable, the system explicitly reports feature unavailability while core screening proceeds without disruption.

To evaluate the pipeline without fabricating real-world recruitment outcomes, we construct a fully reproducible, deterministic synthetic evaluation harness (random.seed(42)). The synthetic corpus contains 100 resumes stratified across 8 technical roles, 4 seniority tiers, and 5 structural layouts, paired against 20 job descriptions spanning four skill-density categories. Through controlled experimentation, we benchmark information extraction, resume–job description matching, multi-format consistency, baseline matching strategies, and structural failure modes.

A. Contributions

The primary contributions of this paper are:

- 1) An Open, Explainable Recruitment Baseline: We document a modular, transparent screening system that maps unstructured resumes to structured entities using a deterministic 108-skill taxonomy and a 100-point linear heuristic scoring rubric.
- 2) Decoupled Architecture with Graceful Degradation: We demonstrate how to integrate optional generative LLM feedback without making the core analytical pipeline dependent on external, stochastic model endpoints.
- 3) A Controlled Synthetic Benchmark Suite: We provide a reproducible, schema-validated synthetic dataset (100 resumes, 20 job descriptions, 320 pairs, 20 cohorts) and an automated evaluation runner.
- 4) Empirical Failure Mode Identification: We quantify concrete failure modes of rule-based information extraction, documenting a 0.0% recall rate when date and title layouts deviate from single-line assumptions, and total section collapse under non-standard headings.
- 5) A Three-Tier Baseline Comparison: We empirically compare strict case-sensitive matching, case-insensitive normalization, and transparent synonym-mapped extraction across 100 resumes.

II. LITERATURE REVIEW

Automated recruitment systems draw on document segmentation, information extraction, semantic matching, and algorithmic fairness.

A. Document Segmentation and Structural Resume Parsing

Resumes are semi-structured documents characterized by variable fonts, multi-column layouts, and heterogeneous section headings. Early work by Yu, Guan, and Zhou [8] demonstrated that "flat" sequence labeling models fail when applied across entire resumes without structural awareness. They proposed a two-pass cascaded hybrid architecture: the first pass segments the document into functional blocks (e.g., *Personal Information*, *Work Experience*, *Education*), and the second pass executes specialized Hidden Markov Models (HMMs) inside those bounded scopes.

Contemporary studies confirm the enduring relevance of this two-pass paradigm. Sajid et al. [7] observed that modern e-recruitment pipelines frequently rely on structural section identification to constrain entity extraction search spaces. For the present lightweight prototype, we therefore retain an explicit text-structure-based parsing approach rather than introducing additional visual-document modeling complexity.

B. Skill Identification and Taxonomy Mapping

Skill extraction serves as a primary bridge connecting job seeker qualifications with employer requirements. Khaouja, Kassou, and Ghogho [3] conducted a systematic survey of 108 studies in *IEEE Access*, classifying skill discovery methods into dictionary-based lookup, supervised sequence taggers, and ontology-guided mapping (such as ESCO and O*NET). They identified dictionary-based matching as a transparent and computationally efficient method, but emphasized its vulnerability to spelling variants, abbreviations, and out-of-vocabulary terms.

To address vocabulary rigidity, Zhang et al. [9] developed the **SkillSpan** benchmark (NAACL 2022), evaluating span-level hard and soft skill extraction across 14.5K job posting sentences. Their experiments showed that domain-adapted models outperform non-adapted counterparts, highlighting the value of domain-specific modeling for skill extraction. Concurrently, Gugnani and Misra [2] investigated implicit skill extraction at AAI/IAAI 2020 using document embeddings (Doc2Vec). Their work illustrates how explicit keyword matching can miss competencies that are implied by job-description context. However, embedding-based inference can also introduce the risk of attributing unverified skills to candidates.

C. Person-Job Fit and Algorithmic Screening

Algorithmic matching between candidate resumes and job postings—termed "Person-Job Fit"—has advanced from Boolean keyword search to representation learning. Qin et al. [5] introduced an ability-aware neural network framework at SIGIR 2018, leveraging dual convolutional and recurrent neural pathways with attention mechanisms to model candidate work experiences against job responsibilities. Similarly, Zhu et al. [10] modeled career trajectories to adapt talent recommendations using historical recruiter hiring decisions.

While continuous representation models capture nuanced semantic relationships, their decision boundaries can be challenging to audit. Raghavan et al. [6] examined commercial algorithmic hiring assessments and discussed how choices concerning prediction targets, training data, validation procedures, and fairness claims can introduce technical and legal risks. Their findings underscore the necessity of transparent, auditable matching algorithms whose criteria can be scrutinized by all stakeholders.

III. RESEARCH GAP AND RESEARCH QUESTIONS

A. Research Gap

The reviewed literature suggests an opportunity to investigate lightweight, open, and reproducible prototypes that combine deterministic information extraction, transparent scoring, optional LLM assistance, and explicit robustness evaluation within a single pipeline. While state-of-the-art neural architectures pursue continuous semantic embeddings, there remains a distinct need for transparent, reproducible baseline systems that can be executed on modest computational hardware, audited line-by-line, and evaluated against controlled structural failure modes.

Our system directly addresses this gap by formalizing a modular rule-based screening architecture evaluated against a controlled synthetic benchmark.

B. Research Questions

To evaluate the system rigorously, we investigate five concrete research questions:

- 1) RQ1 (Information Extraction Accuracy): How accurately can the deterministic rule-based pipeline extract structured information (skills, education, experience, projects, certifications) from semi-structured resumes under controlled synthetic conditions?
- 2) RQ2 (Matching and Gap Detection Effectiveness): How effectively does the set-theoretic matcher identify overlapping technical skills and detect missing job description requirements on synthetic pairs?
- 3) RQ3 (Formatting and Layout Robustness): How robust is the system to variations in document file formats (PDF vs. DOCX vs. plain text), non-standard section headings, and line-break layout styling?
- 4) RQ4 (Baseline Comparison): How does the production case-insensitive skill extraction baseline compare against strict case-sensitive matching and synonym-normalized dictionary mapping?
- 5) RQ5 (Heuristic Score Validity): How closely does the deterministic, weighted 100-point heuristic ATS-style score correlate with the synthetic gold-standard rubric evaluations and cohort rankings?

IV. SYSTEM METHODOLOGY

A. Architecture Overview

The system follows a sequential, modular architecture depicted below:

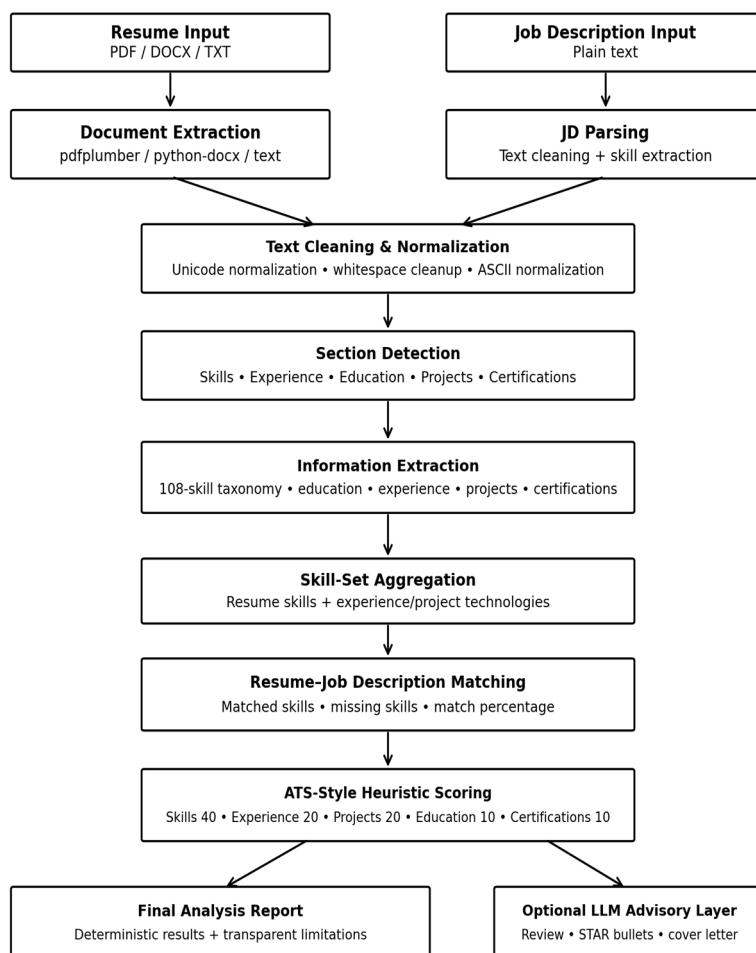


Fig. 1. System architecture of the proposed resume–job description matching system.

B. Document Extraction and Text Cleaning

Resumes submitted as PDF files are processed using pdfplumber, which extracts layout-aware page text streams. Resumes submitted as DOCX files are ingested using python-docx, iterating through paragraph elements and table rows. Plain-text resumes are read directly as UTF-8 text and passed to the same cleaning and section-detection pipeline. The raw text is passed to `clean_text`, which applies Unicode normalization (NFKD), replaces non-breaking spaces and fancy quotation marks with standard ASCII equivalents, removes non-printable control characters, and collapses redundant consecutive whitespaces.

C. Section Detection

Following the cascaded segmentation paradigm [8], `detect_sections` partitions cleaned resume text into functional zones prior to entity extraction. The detector applies a dictionary of normalized regular expression patterns (`SECTION_PATTERNS`) targeting major resume headings:

- Summary: summary, professional summary, profile, objective, about me
- Skills: skills, technical skills, technical expertise, core competencies, expertise
- Experience: experience, work experience, professional experience, employment history, internships
- Education: education, academic qualification, academic background, academics
- Projects: projects, academic projects, personal projects, key projects
- Certifications: certifications, certificates, licenses, courses

Lines matching these section headers delimit text boundaries; all subsequent lines up to the next recognized header are assigned to that functional section. Unmatched preceding text is captured as the document header.

D. Information Extraction

Within each bounded section, specialized deterministic extractors identify domain entities:

1) Skill Extraction

Skill extraction relies on an auditable taxonomy of 108 technical and professional skills organized across 10 functional categories in skills.json:

- Programming Languages (e.g., Python, Java, TypeScript, Go)
- Frontend (e.g., React, Next.js, Tailwind CSS)
- Backend (e.g., FastAPI, Node.js, PostgreSQL)
- Database (e.g., MongoDB, Redis, MySQL)
- AI & ML (e.g., PyTorch, TensorFlow, Scikit-learn, Pandas)
- Cloud & DevOps (e.g., AWS, Docker, Kubernetes, CI/CD)
- BI & Analytics (e.g., Power BI, Tableau, Excel)
- Tools & IDEs (e.g., Git, GitHub, VS Code)
- Testing & Quality (e.g., PyTest, Selenium, Postman)
- Soft Skills (e.g., Problem Solving, Leadership, Teamwork)

Extraction executes word-boundary regular expressions:

Pattern(s) = word-boundary + re.escape(s) + word-boundary.

In the baseline production pipeline, matching is case-insensitive (re.IGNORECASE). A global skill merger (merge_resume_skills) subsequently aggregates skills discovered across the dedicated *Skills* section, *Experience* bullet points, and *Projects* descriptions into a unified candidate skill set:

$$S_{resume} = S_{section} \cup S_{experience} \cup S_{projects}$$

2) Education Extraction

Education details are parsed from the *Education* section using regex patterns matching predefined canonical degree designations:

Degree set $D = \{B.Tech, Bachelor\ of\ Technology, B.E, BCA, MCA, M.Tech, MBA, B.Sc, M.Sc, PhD, Diploma\}$.

CGPA/GPA values are extracted using regular-expression patterns; durations are recognized from year ranges such as 2022–2024, and institutions are identified through keyword matching (University, Institute, College, School).

3) Experience Extraction

The experience parser iterates line-by-line through the *Experience* block. An employment entry is identified when a line contains both a date range matching DATE_PATTERNS (e.g., (Jan|...|Dec)s+\d{4}\s[—]\s(?:Present\d{4})) and an occupational title token matching JOB_KEYWORDS (e.g., engineer, developer, analyst, intern, lead). The title and duration are parsed from the header line, the subsequent non-bullet line is assigned as the company name, and bullet items starting with •, -, or are captured as descriptive accomplishments.

4) Project and Certification Extraction

Project blocks are segmented based on non-bullet header lines preceding bullet descriptions. Technology tokens mentioned in project headers or descriptions are cross-referenced against the skill taxonomy. Certifications are extracted as discrete non-empty lines from the *Certifications* block.

E. Resume–Job Description Skill Matching

The job description parser (parse_job_description) applies text cleaning and extracts required and preferred technical competencies against the identical 108-skill dictionary, yielding the job skill set S_{jd} .

The matcher evaluates compatibility via set operations:

$$S_{matched} = S_{resume} \cap S_{jd}$$

$$S_{missing} = S_{jd} \setminus S_{resume}$$

The match percentage is calculated as:

$$Match\ Percentage = \left(\frac{|S_{matched}|}{|S_{jd}|} \right) \times 100, \text{ when } |S_{jd}| > 0; \text{ otherwise } 100.0$$

F. ATS-Style Heuristic Scoring Engine

The system calculates a transparent, deterministic ATS-style heuristic score bounded in [0, 100]. The overall score is a weighted linear summation of five heuristic components:

$$S_{ATS} = S_{skills} + S_{exp} + S_{proj} + S_{edu} + S_{cert}$$

It is important to emphasize that this score represents a heuristic compatibility index based on keyword and section presence, and is not a validated measure of candidate quality or hiring probability. The mathematical formulation of each component in the codebase is:

1) Skills Score ($S_{skills} \in [0, 40]$):

$$S_{skills} = 40, \text{ when } |S_{jd}| = 0; \text{ otherwise } \text{floor}\left(\frac{|S_{matched}|}{|S_{jd}|} \times 40\right)$$

2) Experience Score ($S_{exp} \in [0, 20]$):

For each parsed experience entry e , points are awarded for a description (+10), extracted technologies (+5), and valid duration (+5), capped at 20:

$$S_{exp} = \min(20, \sum (10 \times I_{desc}(e) + 5 \times I_{tech}(e) + 5 \times I_{dur}(e)))$$

3) Projects Score ($S_{proj} \in [0, 20]$):

For each parsed project p , points are awarded for descriptions (+5) and using ≥ 3 technologies (+5), capped at 20:

$$S_{proj} = \min(20, \sum (5 \times I_{tech \geq 3}(p) + 5 \times I_{desc}(p)))$$

4) Education Score ($S_{edu} \in [0, 10]$):

Points are allocated for degree presence (+4), institution name (+3), and GPA/CGPA mention (+3):

$$S_{edu} = \min(10, 4 \times I_{degree} + 3 \times I_{inst} + 3 \times I_{gpa})$$

5) Certifications Score ($S_{cert} \in [0, 10]$):

Certification score is 10 for three or more certificates, 7 for two, 4 for one, and 0 for none.

G. Decoupled LLM-Assisted Feedback Layer

The system provides three optional generative features powered by Google Gemini (e.g., gemini-2.5-flash):

- 1) Qualitative Resume Review: Generates high-level strengths, growth areas, and alignment recommendations in structured JSON format.
- 2) STAR-Method Bullet Enhancer: Takes an isolated resume bullet point and produces three alternative phrasings incorporating active verbs and quantifiable metrics.
- 3) Tailored Cover Letter Generator: Synthesizes a three-paragraph professional letter aligning candidate background with job requirements.

Graceful Degradation and Failure Transparency: Crucially, the generative LLM is not in the critical path of document parsing, skill extraction, or match scoring. If the GEMINI_API_KEY environment variable is unset, or if API rate limits or network errors occur, `ai_review_unavailable()` returns a structured response:

When Gemini is unavailable, the system explicitly reports an unavailable status. The deterministic extraction, matching, and heuristic scoring pipeline continues without fabricated LLM output.

H. Batch Ranking (Candidate Leaderboard)

For batch candidate evaluation, the `/batch-analyze` endpoint iterates across multiple uploaded resumes against a common job description. Candidates are evaluated deterministically and sorted in descending order of their heuristic ATS-style score S_{ATS} . Ties are resolved deterministically by file arrival order.

V. EXPERIMENTAL SETUP

A. Synthetic Benchmark Dataset

Because real applicant resumes contain sensitive Personally Identifiable Information (PII) and real recruiter hiring records are private, deploying an unverified pipeline on human subjects presents ethical and privacy concerns. Therefore, we developed a deterministic synthetic benchmark (evaluation/synthetic/generator.py, random seed = 42).

The synthetic corpus comprises:

- 100 Resumes: Uniformly distributed across 8 core roles (Backend, Frontend, Full Stack, Data Analyst, Data Science, DevOps, QA, Software Engineer) and 4 experience levels (Student/intern, Junior, Mid-level, Senior).
- 5 Structural Heading Variations: Standard, supported alternatives, unsupported headings, missing sections (omitting education or projects), and scrambled section ordering.
- Controlled Skill Surface Forms: Deliberately injected casing variations (python, FASTAPI), punctuation deviations (Node js, Vue js), informal technology aliases (Postgres, Golang, K8s), non-dictionary skills (Svelte, Julia), skills mentioned exclusively inside projects, and aspirational learning context ("Currently learning Go").
- 20 Job Descriptions: Categorized into 6 high-density (≥ 10 skills), 8 medium-density (5–9 skills), 4 low-density (1–4 skills), and 2 zero-skill non-technical roles.
- 320 Resume–JD Pairs: Formed by pairing each of the 20 JDs with 16 candidates spanning high-fit, partial-fit, low-fit, and zero-fit tiers.
- Synthetic Gold-Standard Derivation: Gold labels for extraction, matching, relevance scores (0–100 rubric), and cohort rankings were mathematically computed directly from generator specifications prior to executing the production pipeline, preventing circular evaluation. These gold labels reflect the generator's internal generative rules and do not represent human expert or recruiter judgments.

B. Evaluation Metrics

Standard information retrieval metrics quantify performance:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}, \quad F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

For score validity across 320 pairs against the synthetic gold-standard rubric, we report Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Pearson linear correlation (r), and Spearman rank correlation (ρ). For cohort rankings, we report Mean Spearman Rank Correlation (ρ) across the 20 distinct candidate groups.

C. Baseline Configurations

To isolate the contribution of text normalization, we benchmark three distinct matching strategies:

- Baseline A (Strict Exact Dictionary): Case-sensitive matching requiring exact casing and punctuation (`\b<Skill>\b`).
- Baseline B (Normalized Case-Insensitive): The production system utilizing case-insensitive regex lookups (`re.IGNORECASE`).
- Baseline C (Synonym-Normalized Mapping): Case-insensitive matching augmented by a transparent dictionary mapping table resolving common aliases (Postgres $\rightarrow$ PostgreSQL, Golang $\rightarrow$ Go, K8s $\rightarrow$ Kubernetes, Node js $\rightarrow$ Node.js).

VI. RESULTS AND DISCUSSION

A. Information Extraction Performance (RQ1)

Table 1 summarizes extraction performance across all 100 synthetic resumes.

TABLE I: INFORMATION EXTRACTION METRICS ON 100 SYNTHETIC RESUMES.

Entity Field	Precision	Recall	F1-Score	True Positives	False Positives	False Negatives
Skills	0.829	0.757	0.791	573	118	184
Education	1.000	0.706	0.827	230	0	96
Experience	0.988	0.634	0.772	755	9	436
Projects	0.768	0.750	0.759	523	158	174
Certifications	0.266	0.797	0.398	47	130	12

Addressing **RQ1**, under the synthetic test conditions, the system achieves high extraction precision for Education (1.000) and Experience (0.988). Education precision is high because the degree matcher evaluates only predefined canonical degree designations. However, its recall is suppressed (0.706) when resumes present non-canonical degrees (e.g., BS in Software Engineering).

Experience precision is high, but recall is constrained to 0.634 due to line-break formatting sensitivity (discussed in Section 7). Certifications exhibit poor precision (0.266) despite acceptable recall (0.797). The extractor ingests any non-empty line in the *Certifications* block exceeding three characters, mistakenly capturing section notes such as "None." and bullet delimiters as credential titles.

B. Resume-JD Matching and Gap Detection (RQ2)

Table 2 details skill matching performance across 320 resume-job description pairs.

TABLE II: RESUME-JOB DESCRIPTION MATCHING METRICS ACROSS 320 SYNTHETIC EVALUATION PAIRS.

Evaluation Target	Precision	Recall	F1-Score	Operational Interpretation
Matched Skills	0.699	0.744	0.721	Correctly identified common competencies
Missing Skills	0.778	0.973	0.865	Correctly identified candidate qualification gaps

Addressing **RQ2**, under the synthetic benchmark, deterministic set subtraction achieved high recall (0.973) and an F1-score of 0.865 for identifying missing recognized skills. This demonstrates the consistency of set-theoretic difference over a closed dictionary in controlled settings, though this synthetic metric reflects dictionary-level tracking rather than real-world candidate qualification assessment. Matched skills exhibit lower precision (0.699) because candidates who list skills in aspirational or learning context (e.g., "exploring Go") are credited with possessing those skills by the uncontextualized regex matcher.

C. ATS Heuristic Score Validity and Cohort Ranking (RQ5)

Table 3 summarizes scoring metrics across the 320 pairs and 20 ranking cohorts.

Table Iii: Score Validity And Cohort Ranking Agreement Against Synthetic Gold-Standard Rubric.

Metric	Empirical Value	Context / Benchmark Interpretation
Evaluated Pairs	320 pairs	Complete evaluation across 20 JDs $\times$ 16 candidates
Mean Absolute Error (MAE)	26.17 / 100	Average divergence between ATS score and synthetic gold-standard rubric
Root Mean Squared Error (RMSE)	33.84	Penalizes large score discrepancies
Pearson Linear Correlation (r)	0.178	Low linear agreement between heuristic score and synthetic gold rubric
Spearman Rank Correlation (ρ)	0.234	Weak monotonic association across all pairs
Cohort Ranking Mean Spearman ($\bar{\rho}$)	0.272	Candidate rank ordering agreement within 20 cohorts

Addressing RQ5, the heuristic ATS-style score showed weak agreement with the synthetic gold-standard rubric ($r = 0.178$, $\rho = 0.234$, cohort $\bar{\rho} = 0.272$), alongside a substantial Mean Absolute Error of 26.17 points.

This result demonstrates that this particular heuristic score does not closely reproduce the synthetic holistic rubric used in this controlled benchmark. The finding underscores that simple linear weighted heuristics of keyword counts cannot be assumed to proxy multi-factor holistic rubrics. While useful as a transparent indicator of keyword overlap, heuristic ATS scores should not be treated as measures of applicant quality or hiring suitability.

D. Baseline Comparisons (RQ4)

Table 4 compares the three rule-based matching strategies across all 100 resumes.

Table IV: Comparison Of Skill Extraction Baseline Strategies Across 100 Resumes.

Baseline Strategy	Precision	Recall	F1-Score	Key Mechanism
Baseline A: Strict Exact Dictionary	0.868	0.896	0.882	Case-sensitive word-boundary matching
Baseline B: Case-Insensitive Normalized	0.791	0.955	0.865	Production baseline (re.IGNORECASE)
Baseline C: Synonym-Normalized Mapping	0.793	0.968	0.872	Case-insensitive + alias mapping table

Addressing RQ4, Baseline A achieves the highest precision (0.868) because strict casing prevents partial-word and casing false positives, but it misses lowercase and uppercase variations. Baseline B (production) improves recall to 0.955 by accepting casing variations, but incurs a precision drop (0.791). Baseline C produced the highest recall (0.968) among the three evaluated strategies under the controlled synthetic benchmark and an improved F1-score (0.872) by mapping informal aliases (Postgres, Golang, K8s, Node js) to their canonical taxonomy entries, demonstrating the value of transparent synonym dictionaries in controlled environments.

E. Multi-Format Extraction Consistency (RQ3)

We evaluated extraction consistency across 25 resumes rendered in parallel as Plain Text (.txt), PDF (.pdf), and DOCX (.docx). Both PDF-to-Text and DOCX-to-Text achieved an identical skill overlap F1-score of **0.800**, 100% agreement on extracted degree designations (25/25), and 100% parity on experience entry counts (45/45). No extraction discrepancies were observed across the tested clean, single-column synthetic documents. However, this finding is specific to the controlled synthetic test suite of 25 single-column documents and should not be generalized to arbitrary, complex real-world PDFs or multi-column DOCX files.

VII. FAILURE ANALYSIS

Through controlled synthetic benchmarking, we identified and quantified five structural failure modes in the rule-based pipeline:

A. Experience Duration–Title Line Binding Failure

- Code Location: backend/app/extractors/experience.py, lines 110–149.
- Empirical Measurement:
- Resumes with Same-Line Format (Software Engineer Jan 2022 - Present): 79.6% recall (113 of 142 entries detected).
- Resumes with Next-Line Format (Software Engineer \n Jan 2022 - Present): 0.0% recall (0 of 28 entries detected).
- Mechanism: The regular expression enforces that a line must simultaneously contain both a valid date pattern (extract_duration) and an occupational title token (looks_like_job_title). When candidates format their resumes with the date on a separate line below the title, the condition evaluates to False on both lines, causing the employment entry to be omitted. The 0.0% recall rate applies specifically to this separated layout format.

B. Section Heading Vulnerability (Section Detection Collapse)

- Code Location: `backend/app/parser/section_detector.py`, lines 3–60.
- Empirical Measurement:
- Standard Headings (Skills, Experience): Skills F1 = 0.869
- Supported Alternative Headings (Technical Expertise, Work Experience): Skills F1 = 0.878
- Unsupported Headings (Core Competencies & Toolkit, Career Trajectory): Skills F1 = 0.000 (0 TP, 0 FP, 184 FN).
- Mechanism: The section detector uses a closed regular expression dictionary. When a resume uses non-standard headings, the detector fails to identify the section boundaries, categorizing the entire text as unassigned header content. Downstream extractors receive empty strings for skills, experience, and education, causing complete extraction failure.

C. Punctuation and Skill Alias Misses

In Baseline B (production), tokens like Node js, Vue js, Next js, and Postgres are missed because the word-boundary regex requires literal punctuation (`\bNode\.js\b`). Baseline C successfully recovers these via its transparent synonym mapping table, illustrating that static regexes without alias resolution penalize candidates who omit punctuation.

D. Zero-Skill Job Description Heuristic Fallback

- Code Location: `backend/app/ats/skills_score.py`, lines 26–27.
- Mechanism: When a job description contains zero recognized skills (e.g., non-technical executive roles in `syn_jd_19` and `syn_jd_20`), the scorer hits a division-by-zero check:

Specifically, when no skills are detected in the job description, the scorer triggers the fallback condition: `if not jd_skills: return 40, [], []`. The system awards full credit (40.0 / 40.0 points) as a fallback. Consequently, candidates receive inflated ATS-style scores (up to 75.0) when matched against non-technical roles. This represents an important edge-case limitation of the current heuristic scoring design.

E. Aspirational Skill False Positives

Candidates who write statements like "Currently exploring and learning C++" are credited with possessing C++ because the uncontextualized regex (`\bC\+\+\b`) matches without semantic dependency parsing. The deterministic regex matcher is context-insensitive: it does not distinguish between verified commercial experience, current proficiency, learning, or aspirational skills.

VIII. ETHICAL CONSIDERATIONS

Algorithmic recruitment tools operate in a domain that directly affects individual livelihoods and employment opportunities. We highlight several ethical considerations governing this work:

- 1) Protection of Candidate Privacy: Real resumes contain sensitive Personally Identifiable Information (PII), contact data, educational histories, and employment trajectories. To avoid handling, storing, or potentially exposing real candidate PII during initial system development, the present evaluation relies exclusively on deterministically generated synthetic documents.
- 2) Non-Autonomous Positioning: The system presented herein is an experimental academic prototype designed for transparent matching analysis. It must not be interpreted or deployed as an autonomous hiring decision-maker.
- 3) Non-Equivalence to Candidate Merit: The heuristic ATS-style score is a rule-based matching index reflecting keyword overlap and section presence; it must not be treated as a measure of candidate competence, intelligence, or hiring suitability.
- 4) Prerequisites for Real-Data Deployment: Any future transition to real-world resume evaluation will require formal ethical review, candidate informed consent, strict data anonymization protocols, and empirical auditing for disparate impact across demographic subgroups. We make no claims regarding demographic fairness or parity in the current implementation.

IX. LIMITATIONS

This research has several explicit limitations:

- 1) Synthetic Evaluation: The benchmark was conducted on synthetic data. While controlled and reproducible, synthetic resumes cannot capture the full typographical chaos, cultural nuances, or narrative complexity of real-world CVs.
- 2) Absence of Human Recruiter Validation: The ATS-style heuristic score was evaluated against a synthetic gold-standard rubric derived from generator rules, not against licensed human recruiters or real-world hiring decisions. No real hiring outcomes were evaluated.

- 3) Fixed Skill Taxonomy: The skill extractor recognizes exactly 108 skills. Unlisted frameworks (e.g., Svelte, Elixir, Prisma) are completely ignored unless manually added to skills.json.
- 4) Section Heading Dependence: Non-standard heading vocabularies cause section detection collapse, as documented in Section 7.2.
- 5) Layout Fragility: Multi-column PDF documents, graphic text boxes, and non-linear visual layouts are vulnerable to text extraction interleaving.
- 6) Context-Insensitive Skill Matching: The system cannot distinguish between verified commercial experience and aspirational self-study.
- 7) Heuristic Scoring Constraints: The linear 100-point scoring model uses static weights that do not dynamically adjust to role-specific hiring priorities.
- 8) Limited Domain Coverage: The skill dictionary and synthetic evaluation focus predominantly on software engineering, data science, and cloud roles, with limited coverage of non-technical disciplines.

X. FUTURE WORK

Future extensions of this codebase should focus on:

- 1) Real-World Anonymized Corpus: Collecting an open, fully anonymized multi-industry resume dataset with inter-annotator agreement metrics (Cohen's κ , ICC) across multiple human recruiters.
- 2) Human Recruiter Evaluation: Benchmarking the matching algorithm and ATS scores against consensus ratings from professional hiring managers.
- 3) Hybrid NER and Contextual Skill Parsing: Augmenting dictionary lookup with lightweight token classifiers and dependency parsing to disambiguate aspirational skills from verified professional experience.
- 4) Ontology Expansion: Integrating standardized labor taxonomies such as European ESCO [1] or O*NET [4] with fuzzy synonym matching.
- 5) Scoring Calibration: Calibrating ATS scoring weights using logistic regression or Bradley-Terry ranking models trained on consensus recruiter preferences.
- 6) Fairness and Demographic Parity Auditing: Evaluating whether section-detection rules systematically disadvantage resumes formatted according to non-Western cultural conventions.

XI. CONCLUSION

In this work, we presented an explainable, rule-based resume-job description matching system that prioritizes algorithmic transparency, computational efficiency, and auditability over black-box predictive claims. By decoupling deterministic extraction from optional generative LLM feedback, the architecture keeps the core deterministic analysis independent of the external LLM service and provides explicit degradation behavior when the LLM service is unavailable.

Benchmarking the pipeline against a controlled synthetic corpus of 100 resumes, 20 job descriptions, 320 evaluation pairs, and 20 cohorts achieved F1-scores of 0.791 for skill extraction and 0.865 for missing-skill identification, while exposing critical failure modes: a 0.0% recall rate on separated date-title lines and complete section collapse under non-standard headings. Furthermore, empirical correlation analysis showed weak agreement ($\rho = 0.272$) between the heuristic ATS-style score and the synthetic gold-standard rubric, highlighting the necessity of caution when deploying automated screening scores.

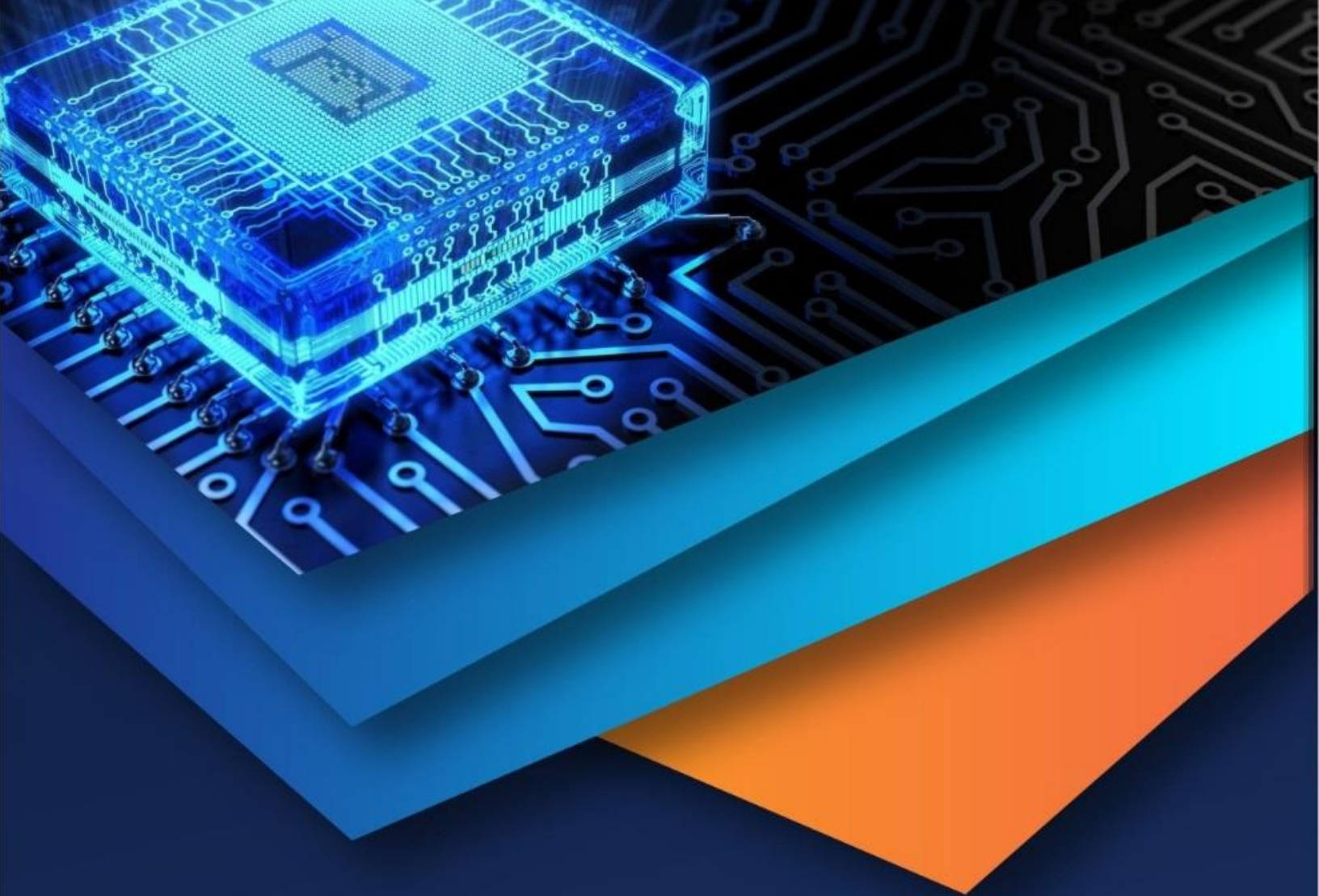
By releasing our system architecture, evaluation harness, and benchmark results openly, we provide a transparent foundation for responsible, verifiable recruitment automation.

REFERENCES

- [1] European Commission. (2022). European Skills, Competences, Qualifications and Occupations (ESCO) Taxonomy v1.1. Directorate-General for Employment, Social Affairs and Inclusion. URL: <https://esco.ec.europa.eu>.
- [2] Gugnani, A. and Misra, H. (2020). Implicit skills extraction using document embedding and its use in job recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '20 / IAAI '20), Vol. 34, No. 08, pp. 13286–13293. DOI: 10.1609/aaai.v34i08.7038.
- [3] Khaouja, I., Kassou, I., and Ghogho, M. (2021). A survey on skill identification from online job ads. IEEE Access, Vol. 9, pp. 118134–118153. DOI: 10.1109/ACCESS.2021.3106120.
- [4] National Center for O*NET Development. (2023). O*NET OnLine: Skills and Knowledge Taxonomies. U.S. Department of Labor, Employment and Training Administration. URL: <https://www.onetonline.org>.
- [5] Qin, C., Zhu, H., Xu, T., Zhu, C., Jiang, L., Chen, E., and Xiong, H. (2018). Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18), pp. 25–34. DOI: 10.1145/3209978.3210025.



- [6] Raghavan, M., Barocas, S., Kleinberg, J., and Levy, K. (2020). Mitigating bias in algorithmic hiring: Evaluating claims and practices. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (ACM FAccT '20), pp. 469–481. DOI: 10.1145/3351095.3372828.
- [7] Sajid, H., Kanwal, J., Bhatti, S. U. R., Qureshi, S. A., Basharat, A., Hussain, S., and Khan, K. U. (2022). Resume parsing framework for e-recruitment. In 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1–8. DOI: 10.1109/IMCOM53663.2022.9721762.
- [8] Yu, K., Guan, G., and Zhou, M. (2005). Resume information extraction with cascaded hybrid model. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05), pp. 499–506. DOI: 10.3115/1219840.1219902.
- [9] Zhang, M., Jensen, K. N., Sonniks, S. D., and Plank, B. (2022). SkillSpan: Hard and soft skill extraction from English job postings. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2022), pp. 4962–4984. DOI: 10.18653/v1/2022.naacl-main.366.
- [10] Zhu, C., Zhu, H., Xiong, H., Zhang, P., and Ding, N. (2018). Person-job fit: Adapting the right talent for the right job with machine learning. ACM Transactions on Management Information Systems (TMIS), Vol. 9, No. 3, Article 12, pp. 1–17. DOI: 10.1145/3234691.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)