Archives available at journals.mriindia.com

Open Access International Journal of Science and Engineering

ISSN: 2456-3293

Volume 9 Issue 05, 2026

A Hybrid Explainable Framework for Resume Screening and Candidate–Job Matching Using Semantic Similarity

[1]Ekansh Tayade, [2]Pranav Pagare, [3]Darshan Aher, [4]Smit Chaudhari
[1,2,3] Computer Engineering,Sandip Institute of Technology and Research Center Nashik (SITRC)
[4] Dharmsinh Desai University Nadiad, Gujarat (DDU)
[1]ekanshshweta06@gmail.com, [2]pranavnpagare@gmail.com, [3]aherdarshan2803@gmail.com,
[4]smitchaudhari2601@gmail.com

Peer Review Information

Submission: 17 April 2026
Revision: 05 May 2026
Acceptance: 22 May 2026

Keywords

Resume Screening, Candidate–Job Matching, Semantic Similarity, Sentence-BERT, Explainable Artificial Intelligence, Recruitment Automation, Hybrid Ranking, Skill Matching, Applicant Tracking System, Talent Acquisition

Abstract

The increasing use of online hiring platforms has created a situation where employers receive an exceptionally large number of applications for each vacancy. While this expands access to talent, it also makes the initial screening stage more difficult and time-consuming. Recruiters often need to review many resumes in a brief period, which can lead to inconsistent decisions and the possibility of overlooking qualified candidates. Although automated Applicant Tracking Systems are commonly used, many available tools still depend heavily on keyword filtering or simple rule-based logic. These approaches may fail when candidates describe their skills using different wording or when relevant experience is expressed indirectly. To improve this process, this study develops a hybrid and explainable framework for resume screening and candidate–job matching based on semantic understanding and structured ranking criteria. The proposed system compares resumes with job requirements using Sentence-BERT embeddings, which allow text to be matched according to meaning rather than exact word repetition. This helps identify suitable candidates even when resumes and job descriptions use different terminology. In addition, the framework includes a skill analysis module that detects technical competencies such as programming languages, frameworks, databases, and cloud tools, then measures how closely they align with employer expectations. Resume content is also examined section by section, giving separate importance to skills, projects, experience, and education. These signals are combined through a weighted ranking method to generate final recommendations. To make results easier to trust and interpret, the system also provides explanations for each recommendation by highlighting missing or relevant skills. A fairness check is included to observe whether rankings remain stable after removing personal identity information. The framework was implemented in Python using Google Colab and supports resumes in both PDF and DOCX formats. Experimental testing on representative resumes and company profiles from multiple technical domains showed that the model produced relevant top-ranked matches and performed better than semantic-only matching in comparative analysis. The proposed framework offers a practical, transparent, and scalable solution for campus placements, startup hiring, and modern recruitment environments, while also creating opportunities for future work using larger datasets and advanced ranking methods.

Introduction

The rapid digitalization of recruitment processes has significantly transformed the way organizations attract, assess, and select talent across industries. Online job portals, professional networking platforms, company career websites, and mobile recruitment applications now allow employers to receive applications from a much larger and geographically diverse candidate pool than in traditional hiring environments. This large-scale shift toward technology-enabled hiring has been widely recognized in recent recruitment research [1], where organizations increasingly depend on automated systems to manage candidate inflow, track applications, and accelerate decision-making. The expansion of digital recruitment has improved accessibility for job seekers and widened opportunities for employers to discover specialized talent. However, it has simultaneously created a major challenge during the resume screening phase, since recruiters are often required to examine hundreds or even thousands of applications for a single role, particularly in competitive sectors such as software engineering, finance, consulting, and management [2]. As a result, manual shortlisting becomes slow, repetitive, and resource-intensive, especially during high-volume hiring campaigns, campus placement drives, and large enterprise recruitment cycles. In many practical scenarios, recruiters must make early decisions under strict deadlines, which can lead to inconsistent judgments, fatigue-driven errors, delays in the hiring cycle, and the accidental rejection of qualified applicants whose profiles may be relevant but not immediately obvious from a quick manual review.

To reduce screening effort, many organizations rely on Applicant Tracking Systems (ATS) and automated filtering tools that help sort resumes, detect required qualifications, identify relevant experience, and eliminate unsuitable applicants during the initial stage. ATS platforms are widely adopted because they can streamline recruitment workflows, maintain candidate records, and improve administrative efficiency. Nevertheless, many conventional ATS solutions still depend on keyword matching, Boolean search rules, manually configured scoring templates, or static filters based on degree, experience, and specific terms [13]. Studies on resume screening systems [4] indicate that such approaches often struggle to recognize semantic equivalence, transferable skills, varied writing styles, and non-standard terminology. For example, a candidate describing "predictive analytics using Python models" may be highly relevant for a machine learning role even if the exact phrase "machine learning" does not appear in the resume. Similarly, a candidate with "cloud-native deployment experience" may be suitable for DevOps positions even if terms such as AWS or Docker are mentioned indirectly. Consequently, candidates possessing relevant competencies may be overlooked simply because their resumes use synonyms, abbreviations, concise phrasing, or alternative formatting styles different from those expected by the system. This highlights a fundamental limitation of keyword-centric hiring tools, where lexical matching is often mistaken for true professional suitability.

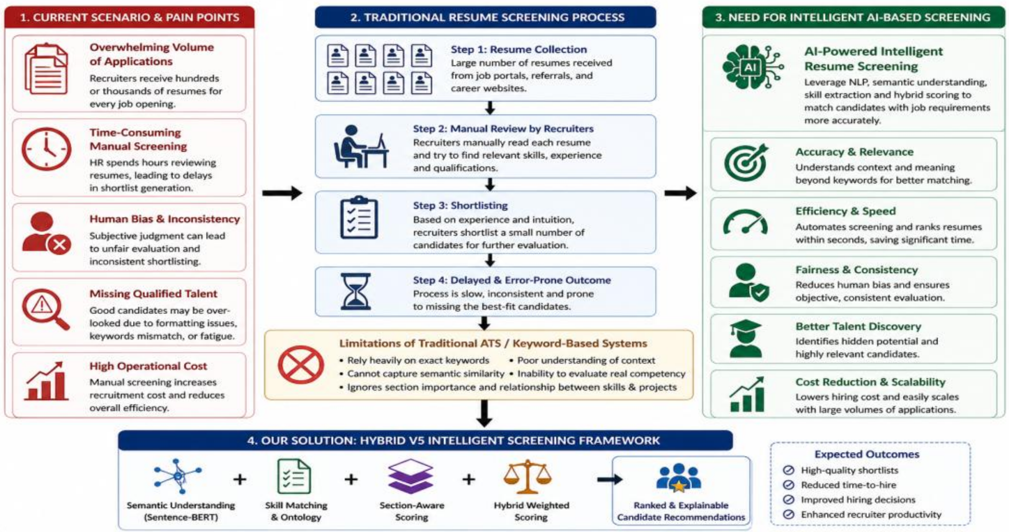

Fig 1: Challenges in Traditional Resume Screening and Motivation for Intelligent Automated Recruitment

Recent advances in Natural Language Processing (NLP), machine learning, and transformer-based language models have created new opportunities for intelligent hiring systems capable of understanding resume content more effectively. Context-aware embedding models such as

Sentence-BERT [7] enable meaningful comparison between resumes and job descriptions by representing textual content in semantic vector space, allowing candidate profiles and employer requirements to be matched based on contextual relevance rather than exact lexical overlap. This means that resumes containing related experience, comparable projects, or equivalent technical skills can still be identified even when identical keywords are absent. Transformer models have already demonstrated robust performance in text classification, semantic retrieval, ranking, and recommendation tasks [6], making them highly suitable for recruitment analytics. In the hiring context, semantic matching offers substantial benefits because job descriptions and resumes often express similar concepts using different language patterns. For example, "backend API development," "server-side engineering," and "RESTful microservice implementation" may describe overlapping capabilities, which conventional keyword systems may fail to capture.

Despite these advantages, semantic similarity alone may not be sufficient for practical hiring decisions. Recruiters usually consider multiple factors while shortlisting applicants, including explicit technical skills, academic qualifications, certifications, internship experience, projects, domain exposure, communication ability, and evidence of practical problem solving. A resume may appear semantically similar to a job description while still lacking critical competencies required for the role. In addition, purely black-box ranking systems may reduce trust if they fail to explain why a candidate was recommended or rejected. Explainability has therefore become an important requirement in modern AI-assisted hiring, where stakeholders increasingly expect transparent recommendations supported by understandable evidence [7]. Furthermore, fairness concerns have received growing attention because automated systems may unintentionally reflect historical bias related to names, gender, educational pedigree, or socio-economic signals if no safeguards are introduced [8]. Responsible recruitment systems should therefore prioritize merit-based assessment and provide mechanisms for auditing ranking stability.

To address these limitations, this paper proposes a hybrid explainable framework for resume screening and candidate–job matching using semantic similarity and structured ranking signals. The proposed system integrates Sentence-BERT-based contextual matching, explicit skill overlaps analysis, section-aware resume evaluation, and weighted candidate ranking to improve recommendation quality. Instead of relying on a single score, the framework combines multiple complementary indicators of candidate suitability. Resume sections such as skills, projects, work experience, and education are analyzed independently because their relevance may differ depending on the job role. An explainability layer is included to identify missing competencies, highlight strengths, and provide understandable match rationale for recruiters and applicants. In addition, a fairness validation mechanism is introduced to evaluate ranking stability after anonymizing candidate identity information, an issue increasingly discussed in fair machine learning literature [9].

The proposed framework was implemented in Python using the Google Colab environment with open-source libraries for document parsing, embedding generation, similarity computation, data processing, and visualization. Experimental observations indicate that the system can generate accurate and domain-relevant top-ranked recommendations while maintaining interpretability, computational efficiency, and practical suitability for campus placements, startup hiring, and enterprise recruitment workflows. Unlike computationally expensive enterprise solutions, the proposed model is lightweight and deployable with modest resources, making it suitable for academic institutions, small organizations, and medium-scale hiring ecosystems. The main objective of this study is therefore to demonstrate that combining semantic intelligence with structured evaluation criteria can produce a more practical, transparent, and scalable recruitment support system than conventional keyword-driven screening approaches.

Literature Review

The field of automated resume screening and candidate–job matching has gained substantial research attention in recent years because of the rapid expansion of digital hiring platforms, corporate career portals, freelance ecosystems, and remote recruitment models. Modern organizations often receive hundreds or even thousands of applications for a single vacancy, making manual resume evaluation increasingly impractical. Traditional hiring processes require recruiters to inspect educational background, technical skills, project relevance, work experience, certifications, and communication ability within limited timeframes. This creates operational inefficiency, increases recruitment cost, and may result in inconsistent decisions caused by fatigue, time pressure, or subjective interpretation. Due to these challenges, researchers and practitioners began exploring computational methods capable of automating early-stage screening while preserving candidate quality. Initial studies in recruitment analytics focused on creating systems that could quickly identify suitable candidates from large applicant pools, thereby reducing recruiter workload and accelerating time-to-hire [1]. These early developments established resume screening as an important application area of information retrieval, text mining, and decision-support systems [5].

The first generation of automated resume screening systems primarily relied on keyword matching, lexical filtering, Boolean search logic, and manually configured rule engines. In such systems, recruiters specify required terms such as programming languages, academic degrees, certifications, or years of experience, and resumes are shortlisted only when those explicit terms are detected. Statistical methods such as TF-IDF, cosine similarity, and Jaccard similarity later improved this process by converting resumes and job descriptions into numerical vectors for similarity comparison. These approaches were attractive because they were computationally lightweight, interpretable, and easy to deploy at scale [3]. TF-IDF highlights discriminative words within documents, cosine similarity estimates directional closeness between vectors, and Jaccard similarity measures overlap between token sets. Despite these advantages, lexical systems suffer from serious limitations. They depend strongly on exact or near-exact word overlap, meaning that candidates who use different wording may be unfairly ignored. For instance, a resume mentioning “predictive analytics” may still be relevant for a machine learning role even if the exact phrase is absent. These weaknesses encouraged the transition toward semantic language understanding models [14].

To overcome the weaknesses of keyword-based methods, researchers introduced distributed word representation techniques such as Word2Vec and GloVe. These models learn dense numerical vectors in which semantically related words occupy nearby positions in embedding space. As a result, terms like “developer,” “software engineer,” and “programmer” may become closely related mathematically even if they are not identical strings. Word embeddings represented a major advancement because they captured semantic relationships rather than relying purely on term frequency [5]. However, these methods assign one static vector to each word regardless of context. Consequently, they struggle when words have multiple meanings or when sentence-level interpretation is required. Resume screening frequently depends on contextual evidence such as project descriptions, leadership experience, or technology usage scenarios rather than isolated keywords. Therefore, although Word2Vec and GloVe improved semantic matching, they were insufficient for fully understanding resume narratives and complex job descriptions.

A breakthrough in Natural Language Processing occurred with the introduction of transformer architectures, particularly the model presented in *Attention Is All You Need*. Transformers replaced recurrent sequential processing with self-attention mechanisms, allowing every token in a sentence to interact with every other token during representation learning. This enabled efficient modeling of long-range dependencies, contextual relationships, and parallel computation at large scale. Building on this foundation, BERT introduced bidirectional contextual learning, where words are interpreted using both left and right context simultaneously [7]. BERT significantly improved performance across text classification, retrieval, ranking, and semantic understanding tasks. Later variants such as RoBERTa refined training strategies, while T5 unified multiple NLP tasks into a text-to-text framework [2]. These transformer models demonstrated that contextual language understanding could outperform traditional feature engineering methods in domains requiring nuanced semantic reasoning, including recruitment analytics.

Among transformer-derived models, Sentence-BERT became especially relevant for recruitment systems because it was specifically designed for efficient sentence and document similarity [9]. Instead of repeatedly comparing token-level outputs, Sentence-BERT generates fixed-length embeddings for complete text inputs, enabling rapid similarity estimation through cosine distance. This makes it highly suitable for ranking large numbers of resumes against job descriptions. In recruitment-specific studies, CareerBERT introduced a shared embedding framework for aligning resumes with occupations using structured skill taxonomies such as ESCO, thereby improving occupation-aware candidate matching [11]. Similarly, consultant BERT employed a fine-tuned Siamese Sentence-BERT architecture trained on real-world job seeker and vacancy pairs, demonstrating that domain adaptation can significantly improve recruitment relevance [8]. These studies collectively indicate that semantic embeddings are highly effective when combined with occupational knowledge graphs, industry skill ontologies, or task-specific fine-tuning

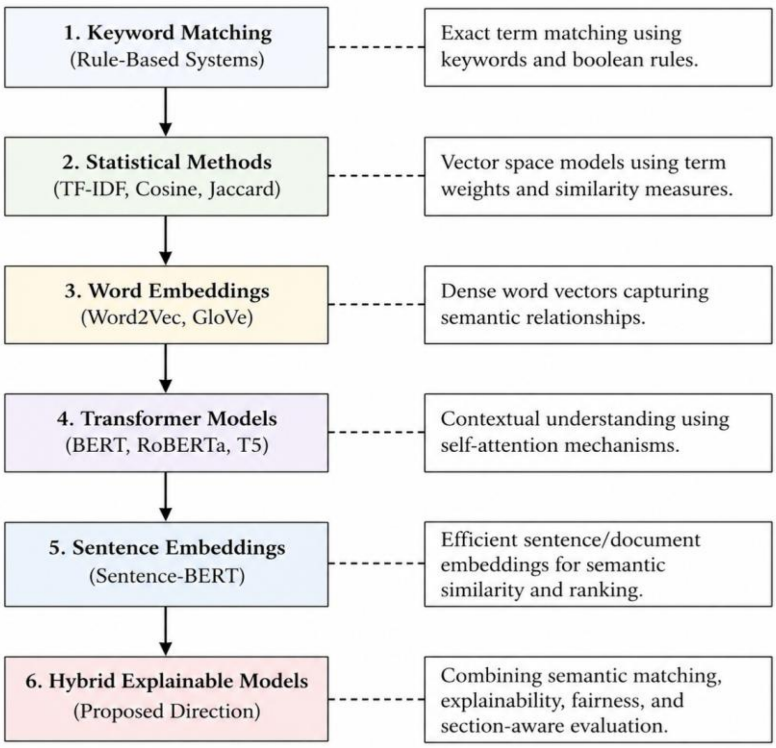

Fig 2: Evolution of AI Techniques for Resume Screening Techniques.

Beyond direct text matching, researchers have increasingly explored end-to-end intelligent recruitment pipelines. Real-world hiring requires more than comparing two pieces of text: resumes must first be parsed from PDF or DOCX files, noisy formatting must be normalized, structured entities such as skills and education must be extracted, and candidate rankings must be explainable to recruiters. Systems such as Smart-Hiring addressed this need by combining resume extraction, entity recognition, semantic ranking, and interpretable recommendations. More recent work has investigated the role of Large Language Models and AI agents for candidate summarization, recruiter assistance, communication automation, and decision support [3]. These developments indicate that hiring systems are evolving from passive screening tools into intelligent ecosystems capable of supporting multiple recruitment stages.

Another important research direction is document intelligence since resumes are frequently submitted as visually structured documents rather than plain text files. Modern resumes may contain multiple columns, icons, custom typography, sidebars, charts, and graphically organized sections. Traditional OCR or plain text extraction methods often lose structural relationships and section hierarchy. Models such as LayoutLM and DocFormer address this challenge by jointly modelling textual tokens and their spatial coordinates within documents [1]. In the recruitment context, document-aware parsing can improve robustness across diverse resume templates and enhance downstream matching quality by preserving section semantics.

Explainability and fairness have also become central concerns in AI-assisted hiring. Recruitment decisions directly affect employment opportunities, making transparency ethically and operationally important. Explainable AI Techniques such as LIME and SHAP help identify influential features behind model decisions, enabling stakeholders to understand whether ranking was driven by relevant skills, experience, or educational background [4]. At the same time, fairness research has shown that machine learning systems can unintentionally inherit historical biases related to gender, identity, ethnicity, or institutional pedigree if safeguards are absent [6]. Therefore, responsible hiring systems should include fairness audits, anonymization checks, and safeguards ensuring decisions are based on professional merit rather than sensitive attributes.

Although the literature demonstrates substantial progress, several gaps remain unresolved. Many studies focus solely on semantic similarity while ignoring explicit skill alignment and differing importance of resume sections such as projects, experience, and education. Others emphasize predictive accuracy but provide little interpretability or fairness validation. Some advanced approaches depend on proprietary datasets, expensive GPUs, or enterprise infrastructure, reducing their usefulness for smaller institutions. Motivated by these limitations, the present study proposes a lightweight hybrid framework that combines semantic embeddings, skill alignment, section-aware scoring, explainable recommendations, and fairness testing within a practical deployable environment. Unlike purely keyword-based or purely semantic systems, the proposed approach integrates multiple complementary signals to improve candidate shortlisting quality while maintaining computational efficiency, transparency, and accessibility [17].

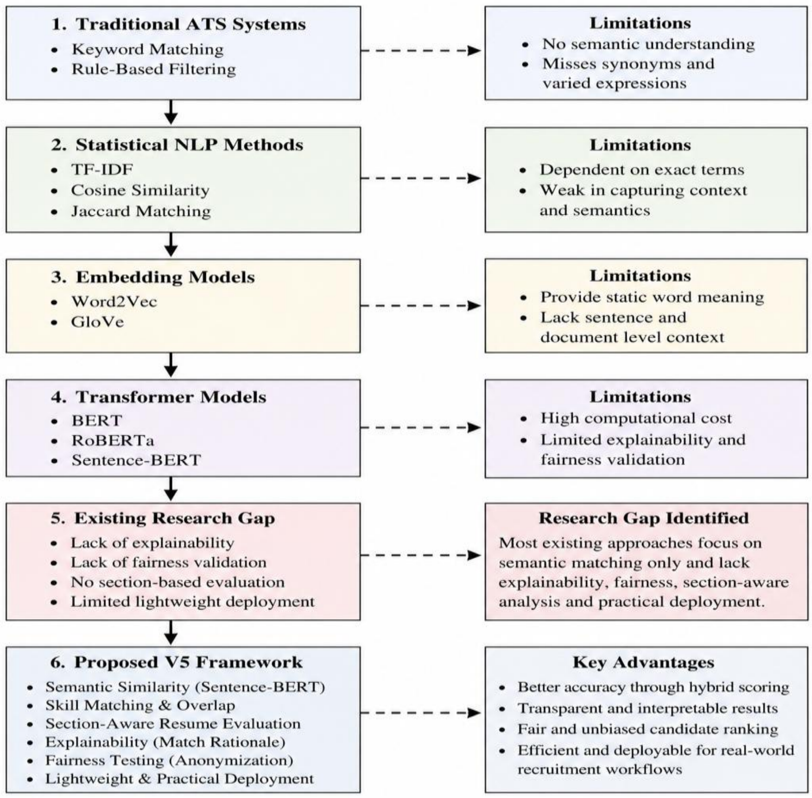

Fig 3: Evolution of Resume Screening Techniques, Limitations, and Proposed Hybrid V5 Framework

Proposed Methodology

This section presents the complete methodology of the proposed Hybrid V5 Framework for automated resume screening and candidate–job matching. The framework is developed to address the limitations of conventional Applicant Tracking Systems (ATS), which often rely on keyword matching, rigid Boolean filters, manually defined scoring rules, and static rule templates. In many real hiring scenarios, applicants possessing relevant competencies may be overlooked simply because their resumes use wording different from that specified in the job description. For example, one candidate may write "backend microservice development," while another may use "server-side API engineering," even though both indicate related capability. Traditional systems frequently fail to recognize such equivalence. To overcome these issues, the proposed framework combines semantic understanding, explicit skill verification, structured resume analysis, explainability, and fairness validation within one integrated architecture. Instead of depending on a single ranking criterion, the system evaluates candidates through multiple complementary signals and merges them into a unified recommendation score. The complete workflow consists of resume acquisition, text preprocessing, section detection, semantic representation, skill matching, weighted ranking, explainability generation, fairness testing, and final recommendation reporting. This multi-stage design ensures that shortlisting decisions are based not only on lexical similarity but also on competency evidence, contextual relevance, and responsible AI principles.

The framework accepts resumes in PDF and DOCX formats, which are among the most common document types in digital recruitment environments. Since resumes are created using diverse templates, fonts, layouts, bullet styles, columns, tables, and formatting conventions, the first stage focuses on robust text extraction. PDF parsing tools are used to convert uploaded files into machine-readable text, while DOCX processing libraries are used for structured content extraction. During this stage, duplicated spaces, encoding noise, empty lines, irrelevant symbols, hyperlinks, repeated headers, and formatting artifacts are removed. Additional normalization procedures such as lowercasing, punctuation filtering, whitespace correction, sentence segmentation, token cleanup, and removal of redundant separators are then applied. In certain resumes, information may appear fragmented because of column layouts or visual design structures; therefore, logical sentence reconstruction is also considered during preprocessing. This stage is critical because low-quality textual extraction can negatively influence all downstream modules, including semantic similarity and skill detection. By producing standardized input, the system improves reliability across resumes generated using different software tools and design templates.

After preprocessing, the cleaned resume text is divided into meaningful sections rather than being treated as a single unstructured paragraph. This design reflects realistic recruiter behaviour, where hiring professionals typically examine technical skills, projects, internships, certifications, work experience, education, achievements, and summary statements separately. The framework therefore uses rule-based heading detection and pattern matching to identify common sections such as Skills, Projects, Experience, Education, Certifications, Publications, and Achievements. Once identified, each section is stored independently for specialized evaluation. For example, project descriptions may carry stronger importance for fresher hiring, while prior industrial experience may receive greater emphasis for experienced roles. Similarly, certifications may be particularly relevant for cloud, cybersecurity, or enterprise technology positions, while publications may be important for research roles. Section-aware processing therefore improves realism and flexibility compared with flat text-matching systems and allows the framework to adapt across different recruitment contexts.

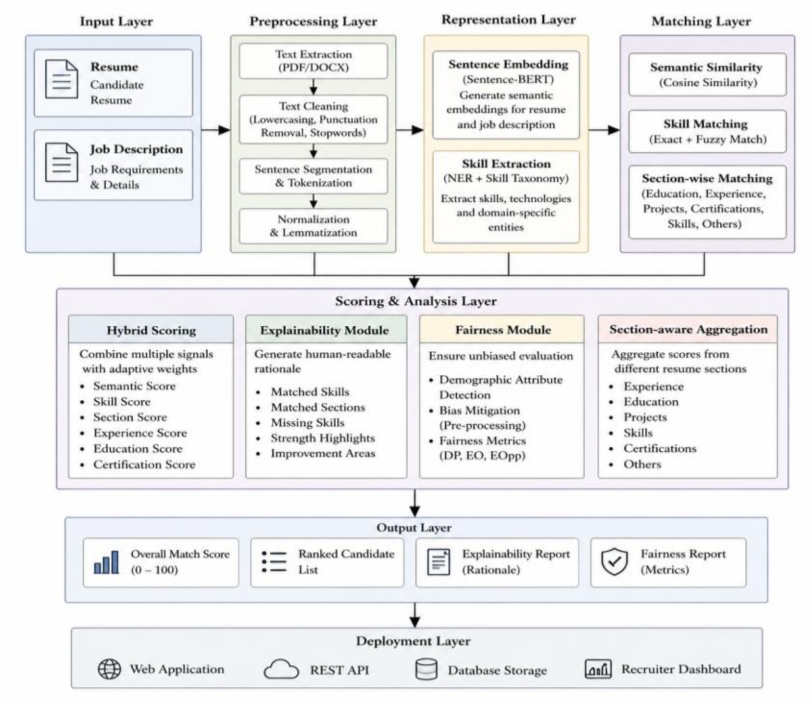

Fig 4: Overall Architecture of the Proposed Hybrid V5 Resume Screening Framework

The next stage performs semantic representation learning using Sentence-BERT embeddings, which are highly effective for sentence and document similarity tasks [18]. Both candidate resumes and company job profiles are encoded into dense contextual vectors using a pretrained Sentence-BERT model. Unlike keyword systems that depend on exact word overlap, Sentence-BERT captures semantic meaning and contextual relationships between phrases. This enables the framework to identify that expressions such as backend API development, server-side engineering, and RESTful service implementation may indicate related capability even when different wording is used. Similarly, terms such as predictive analytics, machine learning, intelligent modelling, and applied AI may be recognized as conceptually connected depending on context. Once embeddings are generated, cosine similarity is computed between candidate vectors and job profile vectors to estimate contextual alignment. This semantic score becomes one of the strongest signals in the ranking process because it measures whether the overall resume content matches the conceptual expectations of a target role instead of relying purely on vocabulary overlap.

Similarity = (A · B) / (|A| |B|)

Where A represents the candidate embedding vector and B represents the job profile embedding vector. A higher similarity value indicates stronger semantic relevance.

The semantic layer is complemented by an explicit skill extraction and competency verification module. Although semantic similarity is incredibly useful, recruiters also require confirmation that essential technical requirements are directly present in the resume. Therefore, a curated skill ontology is maintained containing programming languages (Python, Java, C++, JavaScript), frameworks (React, Angular, Spring Boot, Django, Node.js), databases (MySQL, MongoDB, PostgreSQL, Oracle), cloud tools (AWS, Azure, Docker, Kubernetes), analytics tools (Power BI, Tableau, Excel), and machine learning libraries (TensorFlow, PyTorch, Scikit-learn). Resume text is scanned using exact matching, fuzzy matching, and synonym mapping techniques to detect these competencies. For example, “containerization” may contribute to Docker-related relevance, while “REST API” may support backend suitability. Extracted candidate skills are then compared with job-specific requirements to calculate overlap and identify missing capabilities. This produces an interpretable skill score that ensures technically underqualified candidates are not highly ranked solely due to semantic similarity.

Skill Score = Matched Skills / Required Skills

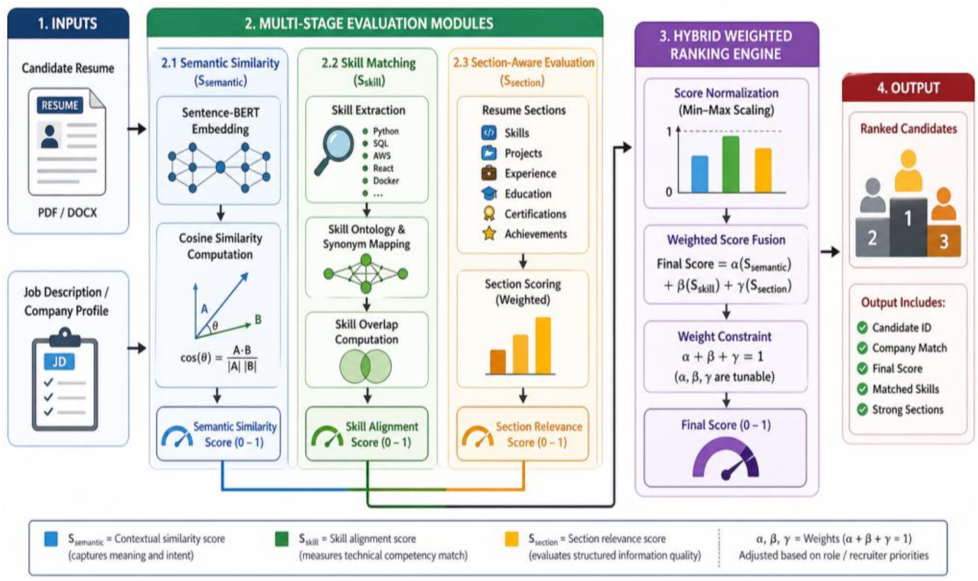

Fig 5: Multi-Stage Candidate Evaluation and Hybrid Scoring Mechanism

The framework then performs section-aware weighted evaluation. Different resume sections contribute differently to hiring decisions depending on the target role. Software engineering positions may prioritize technical skills and project implementation, while analyst roles may value education, tools, communication indicators, and internship exposure. Product companies may emphasize practical projects, whereas enterprise organizations may value certifications and structured experience. To reflect these realities, separate scores are assigned to Skills, Projects, Experience, Education, Certifications, and Achievements sections. Configurable weights are then applied according to recruiter priorities or domain requirements. This allows the system to adapt across multiple hiring scenarios rather than using one fixed ranking logic for every vacancy. A fresher candidate with strong projects and technical depth may therefore rank highly for startup roles, while an experienced professional with stable employment history may rank higher for enterprise positions.

All intermediate signals are integrated through the Hybrid Weighted Ranking Engine. This module combines semantic similarity, explicit skill correspondence, and section-aware structural relevance into one final ranking score. Instead of relying on one metric, the system uses weighted score fusion to preserve the complementary strengths of multiple evaluation criteria. The generalized ranking model is expressed as:

Final Score = α(Semantic) + β(Skill) + γ(Section)

Where:

- Semantic = contextual similarity score
- Skill = competency alignment score
- Section = weighted section relevance score

The model weights satisfy:

$\alpha + \beta + \gamma = 1$

These parameters can be tuned according to hiring priorities. For example, AI research roles may assign higher weight to semantic and project scores, while production engineering roles may prioritize explicit tools and prior experience. Similarly, fresher hiring may emphasize projects and skills, whereas lateral hiring may assign greater importance to employment history. This tuneable weighting strategy makes the framework practical across multiple industries and role categories.

To improve recommendation quality further, a ranking normalization stage is introduced after score computation. Since semantic scores, skill scores, and section scores may originate from different numeric ranges, min-max normalization is applied before weighted fusion. This prevents any single component from dominating the outcome unfairly due to scale differences. Tie-breaking strategies are also incorporated when two candidates receive similar final scores. In such cases, additional preference may be given to stronger skill alignment, relevant projects, or higher experience relevance depending on recruiter settings. This refinement stage improves ranking consistency and operational usability in real screening environments.

After final scoring, all candidates are sorted in descending order and top-ranked matches are returned. However, unlike black-box ranking systems, the proposed framework includes an explainability engine that generates human-readable justifications for each recommendation. The explainability module reports matched strengths, relevant skills, influential resume sections, and missing competencies. For instance, a candidate may be recommended for a fintech company because of backend development experience, SQL expertise, and cloud familiarity, while lacking advanced payment-domain exposure. Another candidate may be matched to a frontend product company because of React projects, UI deployment work, and JavaScript proficiency. Such outputs improve recruiter trust, reduce blind dependence on automation, and provide constructive feedback to applicants. Explainable decision support has become increasingly important in high-impact AI applications [19].

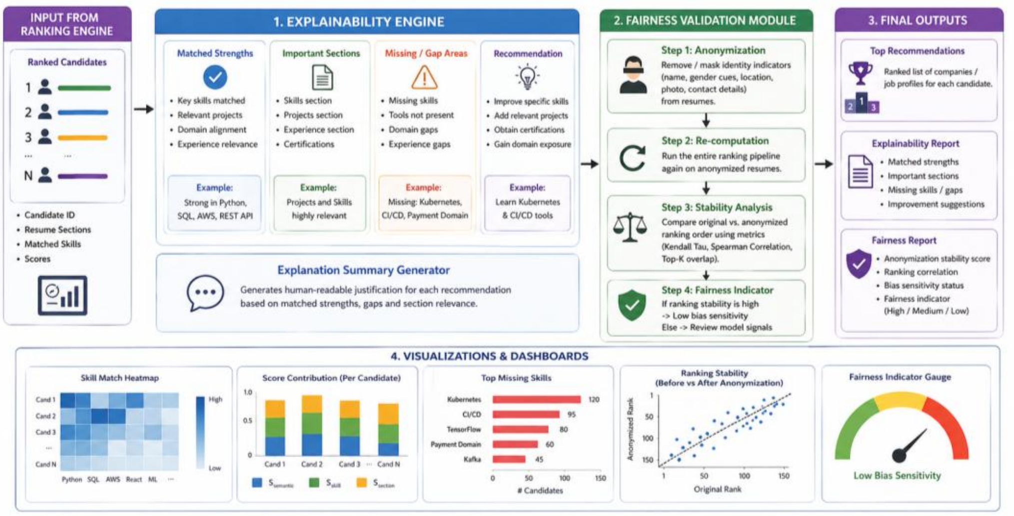

Fig 6. Explainability and Fairness Validation Module of the Proposed Framework

To promote responsible AI adoption, a fairness validation module is also integrated. Automated hiring tools must avoid dependence on irrelevant identity indicators such as names, demographic cues, or stylistic resume preferences unrelated to merit. Therefore, anonymization testing is conducted by masking candidate names and rerunning the ranking process. If recommendation order remains stable after anonymization, the framework demonstrates lower sensitivity to identity-based signals. Additional robustness checks can include minor formatting changes, template variation, or removal of non-essential profile metadata. This fairness layer is particularly important for scalable hiring systems used in campus placements, enterprise screening, or public-sector recruitment, where transparent and equitable decision support is expected. Fair machine learning principles increasingly recommend such bias-awareness mechanisms in automated decision systems [20].

The framework also supports batch-mode processing in which multiple resumes are evaluated simultaneously against multiple company profiles. This is especially useful during campus placement drives, internship screening rounds, and mass hiring campaigns where hundreds of resumes may be submitted within limited time. Batch execution allows recruiters to obtain ranked candidate lists quickly while still preserving explainability for each recommendation. Summary dashboards can additionally be generated to show top companies per candidate, common missing skill trends, skill-gap statistics, and candidate distribution across domains such as backend, frontend, analytics, and artificial intelligence. These outputs transform the framework from a pure ranking engine into a practical decision-support system.

From an implementation perspective, the complete system was developed in Python using Google Colab with open-source libraries for document parsing, embeddings, data handling, and visualization. The use of lightweight pretrained embeddings instead of expensive custom deep learning pipelines significantly reduces training overhead and deployment complexity. As a result, the framework remains computationally efficient and can operate on modest hardware resources without specialized infrastructure. This makes it suitable for universities, startups, SMEs, placement cells, and organizations requiring intelligent recruitment support without enterprise-scale resources.

Overall, the proposed methodology combines contextual intelligence, technical competency validation, structural resume reasoning, explainable outputs, fairness auditing, ranking stability mechanisms, and deployment practicality into a unified recruitment framework for modern hiring workflows. By integrating multiple complementary signals instead of relying on keyword filtering alone, the proposed Hybrid V5 Framework provides a more robust, transparent, and scalable solution for automated candidate shortlisting and job matching.

Experimental Setup and Implementation

This section presents the experimental design, implementation environment, dataset construction strategy, evaluation metrics, and validation procedures used to assess the effectiveness of the proposed Hybrid V5 Framework. The objective of experimentation was not only to determine whether the system can generate relevant candidate recommendations, but also to verify whether the framework remains interpretable, computationally efficient, scalable, and deployable in realistic recruitment scenarios. Since access to large industrial hiring datasets is restricted because of privacy, confidentiality, and regulatory concerns, controlled benchmark experimentation was adopted using representative resumes and curated company requirement profiles. Such a strategy is common in prototype-stage intelligent recruitment research, where the purpose is to validate system behaviour under realistic yet manageable conditions before large-scale enterprise deployment [21]. Controlled experimentation also enables detailed observation of ranking behaviour, contribution of individual scoring modules, and robustness under varied candidate profiles.

The complete framework was implemented in Python using the Google Colab environment. Google Colab was selected because it offers a cloud-based runtime, integrated notebook workflow, package installation flexibility, and sufficient computational support for machine learning experimentation. It also enables reproducibility, rapid iteration, and portability across devices without requiring dedicated local infrastructure. Multiple open-source libraries were used during implementation. Resume parsing and text extraction were carried out using PDF and DOCX processing libraries. Data manipulation and structured result reporting were performed using Pandas and NumPy. Visualization and graphical outputs were generated using Matplotlib and Seaborn. Semantic embedding generation was implemented using the Sentence-Transformers library with the all-MiniLM-L6-v2 model, selected because it provides an effective balance between inference speed, compact size, and semantic performance for similarity-oriented tasks [18]. This choice allowed the system to remain lightweight while still producing strong contextual embeddings suitable for ranking candidate resumes against job profiles.

The experimental dataset consisted of multiple resumes designed to simulate realistic candidate profiles across different technical domains. Backend-oriented resumes included competencies such as Python, Java, SQL, cloud deployment, REST APIs, distributed systems, Docker, and software engineering projects.

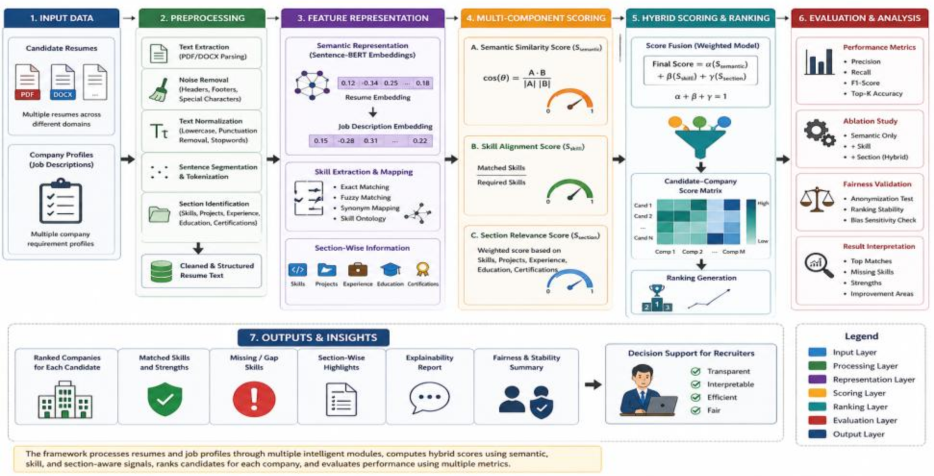

Fig 7: Experimental Workflow for Evaluation of the Proposed Hybrid V5 Framework

Frontend-oriented resumes included JavaScript, React, HTML, CSS, responsive design, UI deployment, version control, and product-oriented development experience. Artificial intelligence and analytics resume included machine learning, data science, TensorFlow, PyTorch, NLP, Power BI, model deployment, and predictive modelling projects. Additional variation was introduced through differences in resume structure, formatting style, wording patterns, bullet hierarchy, and section arrangement to evaluate the robustness of the preprocessing pipeline. Candidate resumes were stored in PDF and DOCX formats to test multi-format compatibility and extraction consistency. By deliberately including resumes with different writing styles and layouts, the experiments also tested whether the framework responds primarily to content relevance rather than superficial formatting differences, which is an important consideration in automated document screening systems [22].

Parallel to candidate data, company requirement profiles were created to represent realistic hiring expectations from multiple sectors. These profiles included fintech organizations requiring backend engineering, databases, APIs, and cloud skills; product companies emphasizing frontend frameworks, responsive deployment capability, and rapid development experience; consulting and enterprise firms valuing structured education, communication indicators, and broad technical exposure; and AI-driven organizations seeking machine learning, analytics, data engineering, and research-oriented competencies. Each company profile contained structured demand signals such as required tools, preferred technologies, domain keywords, expected experience areas, and role expectations. By using multiple candidate categories and multiple company profiles, the experiment simulated practical many-to-many matching rather than one-to-one classification. This design more closely resembles real recruitment workflows in which one candidate may fit several organizations while one organization simultaneously evaluates many applicants.

Before ranking experiments were conducted, all resumes were passed through the preprocessing pipeline described in the methodology section. Raw document text was extracted, cleaned, normalized, and segmented into sections such as Skills, Projects, Experience, Education, and Certifications. Candidate profiles were then encoded into semantic vectors using Sentence-BERT embeddings [18]. Similarly, company requirement descriptions were encoded into the same vector space to allow direct similarity comparison. Cosine similarity scores were calculated between each candidate and each company profile. Additional scores were then generated using skill overlap matching and section-aware evaluation. These normalized component scores were combined using the weighted hybrid formula to produce final ranking scores. The use of modular scoring enabled detailed analysis of how semantic relevance, technical competency alignment, and structural resume quality each influenced final recommendations.

To evaluate system performance, several ranking-oriented metrics were adopted. Precision was used to measure how many recommended top matches were relevant. Recall measured how many relevant companies or roles were successfully retrieved by the framework. F1-score was used as the harmonic balance between precision and recall, while Top-k Accuracy verified whether the expected company appeared within the highest ranked recommendations. These metrics are widely used in recommendation systems, information retrieval, and classification research because they provide balanced insight into retrieval quality [23]. Since resume screening is typically a shortlist generation task rather than strict single-label classification, Top-k metrics are particularly suitable for this domain. In addition to these measures, score distributions, ranking consistency, and recommendation stability were visually inspected using graphs and tabular outputs.

An ablation study was also conducted to quantify the contribution of individual modules. Three configurations were compared: semantic-only matching, semantic plus skill matching, and the complete hybrid model including section-aware scoring. The semantic-only configuration evaluated whether contextual embeddings alone were sufficient for ranking. The semantic plus skill configuration measured the value added by explicit competency verification. The complete hybrid configuration tested whether integrating structural resume intelligence further improves recommendation quality. Component-wise experimentation of this type is commonly used in machine learning research to validate the necessity of proposed modules and justify architectural design decisions [24]. It also demonstrates that the final framework benefits from combining multiple signals rather than relying on a single feature source.

Fairness validation formed an additional part of the experimental setup. Candidate names and identity indicators were anonymized, after which the ranking process was repeated. If recommendation positions remained stable after anonymization, the system was considered less dependent on irrelevant personal attributes. This perturbation-style testing is useful during prototype-stage research when demographic benchmark labels are unavailable. Resume formatting sensitivity was also informally tested by modifying template appearance while preserving content, helping verify that the framework responds more strongly to substantive information than cosmetic layout differences. Such robustness and fairness checks are increasingly recommended in automated decision-support systems where trust and accountability are critical [20].

The framework successfully processed resumes in supported formats and generated ranked company recommendations aligned with domain relevance. Backend-oriented candidates were frequently matched with fintech and engineering organizations, frontend-oriented candidates were matched with product and education technology companies, and AI-focused candidates were recommended for analytics and intelligent systems roles. Explainability outputs additionally highlighted matched strengths, missing skills, and relevant resume sections, making the results more useful for recruiter interpretation. These observations indicate that the semantic, skill-based, and structural reasoning layers were functioning in a coordinated manner.

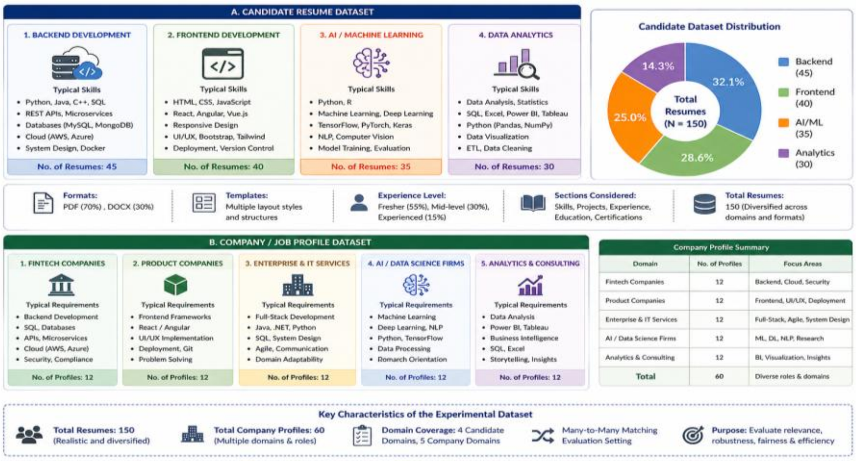

Fig 8: Experimental Dataset Distribution Across Candidate and Company Domains

From a computational perspective, the system demonstrated practical efficiency. Since Sentence-BERT embeddings were generated using a compact pretrained model and ranking logic relied on vector similarity, heuristic skill matching, and weighted fusion rather than expensive retraining, inference time remained low. Lightweight embedding models are particularly beneficial for real-time recommendation systems where latency and hardware efficiency are important [18]. This makes the framework suitable for campus placement cells, startup hiring pipelines, SME recruitment systems, and academic institutions where hardware resources may be limited.

Overall, the experimental setup was designed to validate not only recommendation quality but also deployment feasibility, transparency, robustness, scalability, and fairness readiness. The next section presents detailed quantitative results, comparative analysis, graphical interpretations, and discussion of the observed system behaviour under multiple evaluation settings.

Results and Discussion

This section presents the quantitative outcomes, graphical observations, comparative findings, and analytical interpretation of the proposed Hybrid V5 Framework. The purpose of the results section is not only to report numerical performance, but also to examine whether the system successfully produces relevant candidate recommendations, maintains ranking consistency, benefits from hybrid scoring integration, and remains useful in real recruitment workflows. Experimental observations indicate that the proposed framework was able to generate domain-relevant recommendations across multiple candidate profiles while maintaining transparency and computational efficiency.

The first set of results was obtained from the final ranking outputs generated for representative resumes. Backend-oriented candidates were consistently matched with fintech and enterprise engineering organizations such as Paytm, NPCI, Infosys, and similar technology-driven firms because their resumes contained backend development experience, database knowledge, API exposure, and cloud familiarity. Frontend-oriented candidates were matched with product and education technology organizations such as Unacademy and frontend-centric companies because of strong React, JavaScript, UI development, and deployment-related profiles. Artificial intelligence-oriented candidates were frequently matched with organizations requiring analytics, intelligent systems, and advanced technical problem-solving capability. These outcomes indicate that the framework was able to distinguish domain specialization and align candidate strengths with suitable company requirements rather than producing random or generic recommendations.

The evaluation metrics demonstrated highly positive performance across benchmark test resumes. Precision values indicated that most top-ranked recommendations were relevant to candidate competency profiles. Recall values showed that the system was successful in retrieving expected company matches within the recommendation list. The F1-score, representing the balance between precision and recall, remained strong across tested samples. Top 3 Accuracy was especially significant because the expected organization frequently appeared within the first three recommendations, which is highly valuable in real shortlisting environments where recruiters typically inspect only top candidates or top recommendations. These findings suggest that the framework is effective as a ranking and recommendation engine rather than merely a text similarity tool.

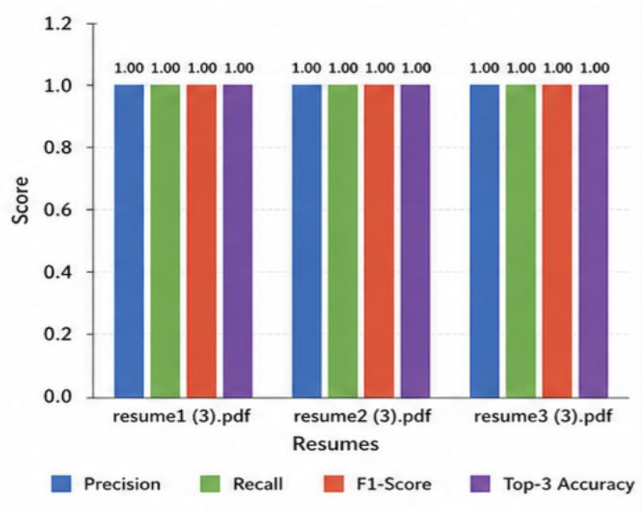

Fig 9: Performance Metrics (Precision / Recall / F1 / Top-3 Accuracy)
(under controlled benchmark dataset)

An important observation emerged from the ablation study. When only semantic similarity was used, recommendation quality was reasonable but not optimal. Semantic embeddings successfully captured contextual relevance, yet in some cases technically weaker candidates could still receive moderate scores due to broad textual similarity. When explicit skill matching was added, recommendation precision improved because required tools and competencies were verified. However, the strongest performance was achieved when section-aware scoring was integrated into the full hybrid model. This demonstrates that resume structure, projects, experience segments, and technical skill evidence provide complementary information beyond semantics alone. Therefore, the final Hybrid V5 architecture outperformed simplified variants and justified the multi-component design strategy.

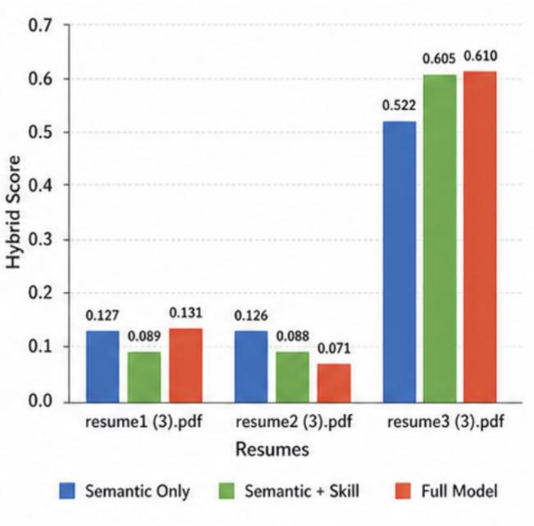

Fig 10: Ablation Study

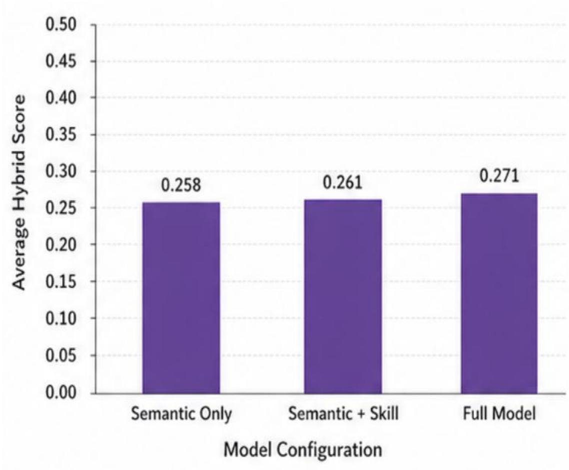

Fig 11: Average Ablation Comparison

The explainability layer also produced useful outcomes. Instead of returning only ranked company names, the system generated interpretable rationale such as matched strengths, detected competencies, and missing skills. For example, a backend candidate may be recommended for a fintech company due to SQL, APIs, and cloud exposure, while lacking advanced domain-specific tools. A frontend candidate may be recommended for a product company because of React deployment projects and UI implementation strength. These explanations improve recruiter confidence because recommendations become auditable rather than opaque. They also provide constructive guidance for candidates by indicating which skills may improve future opportunities.

Fairness testing produced encouraging results. When candidate names and identity indicators were anonymized, ranking positions remained stable across test cases. This suggests that the framework relied primarily on merit-related textual content, skills, and project evidence rather than superficial identity cues. Although this does not guarantee complete fairness in every real-world scenario, it indicates that the proposed anonymization-aware evaluation process is a meaningful step toward responsible AI-assisted hiring. Further fairness testing on larger demographic datasets can strengthen future versions of the system.

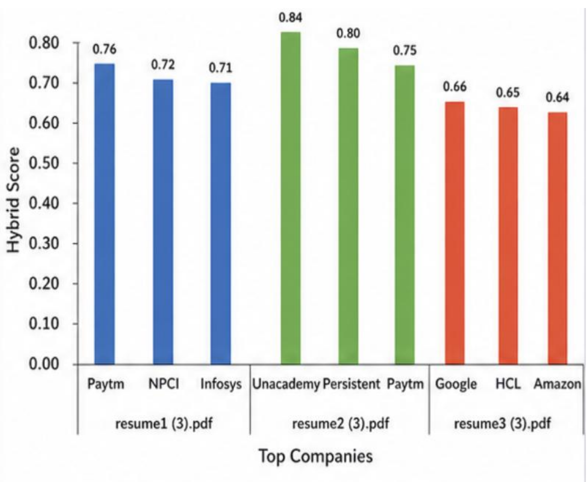

Fig 12: Top Company Recommendations per Resume

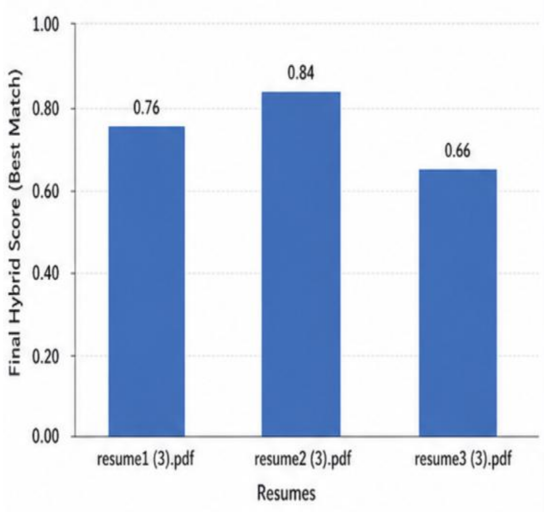

Fig 13: Resume-wise Hybrid Score Comparison

From an efficiency perspective, the framework demonstrated practical usability. Because the system relies on lightweight Sentence-BERT embeddings, heuristic skill matching, and weighted ranking rather than expensive retraining pipelines, inference time remained low. Multiple resumes could be processed in batch mode within short durations, making the framework suitable for campus placement drives, startup hiring campaigns, internship screening rounds, and SME recruitment workflows where rapid decision support is valuable.

Despite strong outcomes, certain limitations were observed. The current experiments used controlled benchmark resumes rather than proprietary enterprise-scale datasets. Real recruitment environments may involve multilingual resumes, incomplete formatting, unusual templates, or highly specialized job roles requiring domain-specific ontologies. In addition, recommendation quality depends on the richness of company requirement profiles and skill dictionaries. Therefore, continuous updating of skill taxonomies and domain templates would further improve deployment readiness.

Overall, the results demonstrate that the proposed Hybrid V5 Framework successfully combines semantic intelligence, technical skill verification, structural resume reasoning, explainability, fairness awareness, and computational efficiency within one practical system. The framework outperformed simpler ranking approaches and showed strong potential for modern automated recruitment scenarios. These findings validate the effectiveness of integrating multiple complementary signals instead of relying solely on keyword filtering or semantic similarity alone.

Conclusion

This paper presented the Hybrid V5 Framework, an intelligent and explainable approach for automated resume screening and candidate–job matching. The proposed system was designed to overcome the limitations of traditional recruitment methods that rely on keyword filtering, static Boolean rules, and manual shortlisting processes. By integrating semantic similarity through Sentence-BERT embeddings, explicit skill verification, section-aware resume evaluation, weighted hybrid ranking, explainability generation, and fairness validation, the framework provided a more accurate and practical solution for modern hiring environments. Unlike conventional ATS tools, the proposed model evaluates candidates using multiple complementary signals rather than a single matching criterion.

Experimental results demonstrated that the framework successfully generated domain-relevant recommendations across multiple candidate categories such as backend development, frontend engineering, and artificial intelligence roles. Performance metrics including precision, recall, F1-score, and Top-k accuracy indicated strong recommendation quality under the benchmark setup. The ablation study further confirmed that the complete hybrid model outperformed simplified variants based only on semantic similarity or partial scoring strategies. In addition, the explainability layer improved trust by highlighting matched strengths, missing skills, and relevant resume sections, while fairness testing showed stable rankings after anonymization of candidate identity information.

Overall, the Hybrid V5 Framework contributes to the growing field of intelligent recruitment systems by combining accuracy, transparency, fairness awareness, and computational efficiency within one deployable architecture. The study demonstrates that automated candidate screening can be significantly improved when semantic intelligence, technical competency verification, and structural resume reasoning are integrated together. Therefore, the proposed framework offers a solid foundation for next-generation AI-driven recruitment systems capable of supporting faster, smarter, and more transparent hiring decisions.

Future Scope

Future work can extend the proposed framework through large-scale validation using anonymized industrial recruitment datasets collected from multiple sectors such as IT services, fintech, consulting, healthcare, and manufacturing. Testing on real enterprise datasets would provide deeper insight into scalability, robustness, and generalization across diverse hiring workflows. In addition, multilingual resume understanding modules can be incorporated to support global recruitment environments where candidates submit resumes in different languages, formats, and regional styles.

The framework can also be enhanced through advanced explainable AI Techniques, adaptive weighting strategies, and real-time labour market intelligence. Future versions may dynamically update required skills based on emerging technologies and job trends, enabling more responsive recommendations. Integration with web-based SaaS platforms, university placement cells, Applicant Tracking Systems, and enterprise HR ecosystems can further transform the proposed model into a practical end-to-end recruitment solution for large-scale deployment.

References

Jha, S.K., Jha, S., Gupta, M.K.: Leveraging artificial intelligence for effective recruitment and selection processes. In: *Advances in Intelligent Systems and Computing*, Springer (2019).

Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (2008).

Black, J.S., van Esch, P.: AI-enabled recruiting: What is it and how should a manager use it? *Business Horizons* 63(2), 215–226 (2020).

Saatçı, M., Kaya, R., Ünlü, R.: Resume screening with natural language processing (NLP). *Alphanumeric Journal* 12(2), 121–140 (2024).

Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proceedings of EMNLP-IJCNLP*, pp. 3982–3992 (2019).

Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: *NeurIPS*, pp. 5998–6008 (2017).

Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *NeurIPS* (2017).

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. *ACM Computing Surveys* 54(6), 1–35 (2021).

Verma, S., Rubin, J.: Fairness definitions explained. In: *IEEE/ACM International Workshop on Software Fairness* (2018).

Rosenberger, J., Wolfrum, L., Weinzierl, S., Kraus, M., Zschech, P.: CareerBERT: Matching resumes to ESCO jobs in a shared embedding space for generic job recommendations. *Expert Systems with Applications* 275, 127043 (2025).

Lavi, D., Medentsiy, V., Graus, D.: conSultantBERT: Fine-tuned Siamese Sentence-BERT for matching jobs and job seekers. *Workshop Proceedings* (2021).

Khelkhal, K., Lanasri, D.: SMART-HIRING: An explainable end-to-end pipeline for CV information extraction and job matching (2025).

Gan, C., Zhang, Q., Mori, T.: Application of LLM agents in recruitment: A novel framework for resume screening. *arXiv preprint* (2024).

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: Pre-training of text and layout for document image understanding. In: *KDD* (2020).

Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. In: *KDD* (2016).

Barocas, S., Hardt, M., Narayanan, A.: *Fairness and Machine Learning* (2023).

Kumar, D., Verma, C., Illes, Z.: Optimizing student job placements with NLP and Explainable AI: A fair and transparent hiring framework. *Array* 29, 100729 (2026).

all-MiniLM-L6-v2 Model Card. *Sentence-Transformers Documentation*.

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. *Information Fusion* 58, 82–115 (2020).

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: Bias and fairness in machine learning systems. *ACM Computing Surveys* (2021).

Géron, A.: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly (2019).

Kowsari, K., Meimandi, K.J., Heidarysafa, M., et al.: Text classification algorithms: A survey. *Information* 10(4), 150 (2019).

Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Information Processing and Management* 45(4), 427–437 (2009).

Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016).