International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Contextual Embedding-Based Resume Screening with Semantic Relevance Scoring and Qualified Candidate Retention

Y. Vijaya Lakshmi[1], Assistant Professor, Department of CSE,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India.

V. Vijay Kumar Yadav[2], P. Sowmya[3], T. Hemima[4], R. Hema Hari Teja Sri[5]
[2,3,4,5]UG Students, Department of CSE,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India.
[1]vijayalakshmiguntamukkala16@gmail.com [2]22bq1a05n0@vvit.net, [3]22bq1a05k7@vvit.net,
[4]22bq1a05l5@vvit.net, [5]23bq5a0520@vvit.net

***Abstract*—Modern hiring platforms face mounting pressure to process large volumes of applications without sacrificing candidate quality. Conventional screening tools built on token-matching logic are structurally incapable of recognizing conceptual alignment between applicant profiles and position requirements, resulting in disproportionate rejection of genuinely qualified individuals.**

This study presents an intelligent recruitment screening system that combines deep contextual text representations, high-speed vector retrieval, and structured competency mapping. A purpose-built relevance annotation workflow is embedded directly into the hiring pipeline to support credible system benchmarking. The platform is constructed around a service-oriented architecture using FastAPI for server-side logic, React for the recruiter interface, and FAISS for vector-based candidate retrieval.

Controlled experiments confirm that the proposed approach substantially outperforms both lexical and static-embedding baselines on ranking quality, retrieval coverage, and candidate retention metrics. Query response times remain below 100 milliseconds, confirming viability for production-scale deployment. The findings underscore the value of deep language understanding in building equitable and efficient AI-assisted hiring tools.

I. Introduction

Digital hiring ecosystems have undergone a profound shift over the past decade. The proliferation of job boards, professional networks, and online application portals has dramatically expanded the candidate pool available to employers. While this expansion benefits organizations seeking diverse talent, it simultaneously creates a bottleneck at the initial screening stage, where human reviewers cannot feasibly evaluate every submission.

Automated screening tools, commonly referred to as Applicant Tracking Systems (ATS), were introduced to manage this volume. Most deployed systems, however, operate on surface-level text matching, comparing job postings and resumes based on shared vocabulary. This design assumes that lexical overlap reliably signals candidate suitability — an assumption that breaks down in practice due to the natural diversity of professional language.

Consider a scenario where a position calls for expertise in ”machine learning” but a strong applicant describes their background using terms like ”predictive analytics” or ”statistical modeling.” A keyword-driven system would likely discard this candidate despite genuine competency alignment. Beyond accuracy concerns, such systems can also disadvantage candidates who are unfamiliar with resume optimization strategies, introducing a form of systemic bias.

Advances in deep learning, particularly the development of transformer-based language models, have opened new possibilities for understanding text at a conceptual level. These architectures use attention mechanisms to model relationships between words in context, producing representations that capture meaning rather than just form. Applied to document comparison, they enable matching based on semantic intent rather than literal word choice.

While transformer models have proven effective in search, question answering, and document classification, their use in automated hiring remains limited. Most recruitment-focused research addresses narrower problems such as resume categorization or job recommendation, and few studies have examined end-to-end semantic matching with rigorous real-world evaluation.

A further gap in the literature concerns evaluation quality. Studies that rely on synthetically generated or heuristically assigned labels cannot reliably reflect how actual hiring professionals assess candidate fit. Without grounded evaluation data, reported performance figures may not translate to operational settings.

This work introduces a recruitment intelligence system that addresses these shortcomings. The system encodes both job descriptions and candidate resumes using contextual sentence-level embeddings, retrieves candidates via approximate vector search, and ranks them using a composite score that incorporates semantic alignment, skill coverage, and experience fit. A structured feedback mechanism allows recruiters to annotate

results, generating authentic evaluation labels.

The primary contributions of this work are:

- A contextual semantic matching pipeline adapted for recruitment screening.
- A structured competency alignment module using domain skill taxonomies.
- A recruiter-driven annotation protocol for generating realistic evaluation data.
- Hiring-specific performance metrics with emphasis on reducing unwarranted rejections.
- A production-ready system architecture enabling real-time candidate retrieval.

The paper proceeds as follows. Section II outlines the research contributions and novelty. Section III surveys related work. Section IV describes the system design. Section V details the matching methodology. Section VI covers the dataset and evaluation setup. Section VII presents experimental results. Section VIII discusses findings and limitations. Section IX concludes the paper.

II. Research Contributions

Semantic similarity research has a long history in information retrieval, but applying it to hiring introduces domain-specific constraints around skill equivalence, ranking fairness, and evaluation credibility. This work tackles these constraints through a unified intelligent recruitment framework.

A central contribution is the formulation of a Qualified Candidate Retention (QCR) objective, which explicitly penalizes the erroneous elimination of suitable applicants. This contrasts with conventional retrieval systems that optimize only for precision or recall without accounting for the cost of rejecting genuinely qualified individuals.

The system also introduces a multi-signal scoring mechanism that blends contextual semantic similarity with structured skill overlap and experience compatibility. This composite approach produces more nuanced relevance estimates than single-signal methods.

To support credible evaluation, a human-annotated labeling workflow is embedded within the hiring interface. Recruiter judgments on ranked candidate lists are captured and stored, enabling the construction of evaluation datasets that reflect real hiring decisions rather than proxy labels.

From an engineering standpoint, the work demonstrates a deployable architecture that connects transformer inference, vector indexing, and a web-based recruiter interface into a cohesive real-time system.

A. Novelty Justification

Existing recruitment AI systems typically frame the problem as classification or recommendation. This work reframes it as a semantic retrieval task, which better captures the open-ended nature of candidate-job alignment and allows the system to generalize across diverse role types without retraining.

The explicit optimization for qualified candidate retention is a domain-specific objective that has received limited attention in the broader information retrieval literature. Combined with the recruiter-in-the-loop annotation mechanism, this work offers a more practically grounded approach to recruitment AI than prior systems.

III. Related Work

Research on automated resume screening spans multiple disciplines including information retrieval, NLP, recommender systems, and HR analytics. This section reviews foundational and recent work relevant to the proposed system.

A. Lexical Retrieval in Hiring Systems

Early automated screening tools were built on classical IR models including the Vector Space Model and BM25. TF-IDF weighting remains common in deployed systems due to its simplicity and low computational cost. These approaches represent documents as sparse term vectors and measure similarity through vocabulary overlap.

The core weakness of lexical methods is their inability to bridge vocabulary gaps. Two documents discussing the same concept using different terminology will appear dissimilar to a keyword-based system. In recruitment, where professional language varies widely across industries and career stages, this limitation is particularly damaging.

B. Word Embedding Approaches

Distributed word representations such as Word2Vec and GloVe improved upon lexical methods by encoding semantic relationships in dense vector spaces. Document-level representations can be constructed by aggregating word vectors, enabling similarity comparisons that go beyond exact term matching.

However, these models assign fixed representations to each word regardless of context. A term like ”python” receives the same vector whether it refers to a programming language or a snake. This context-blindness limits their utility for technical documents where terminology is highly domain-specific.

C. Contextual Language Models

The introduction of BERT and related transformer architectures marked a significant shift in NLP capability. These models generate token representations conditioned on surrounding context, enabling disambiguation and nuanced semantic understanding. Sentence-level variants such as Sentence-BERT produce fixed-length embeddings suitable for efficient similarity computation.

Transformer embeddings have demonstrated strong performance across retrieval, ranking, and matching tasks. Their application to recruitment screening is a natural extension, though it has been explored in limited depth in the literature.

D. AI in Talent Acquisition

Machine learning has been applied to various stages of the hiring process, including resume parsing, candidate scoring, and interview scheduling. Most supervised approaches require labeled training data that is expensive to collect and may encode historical hiring biases.

Recommender system techniques have also been adapted for job-candidate matching, typically using collaborative filtering on historical application and hiring data. These methods struggle with cold-start scenarios and do not generalize well to novel job roles or candidate profiles.

E. Scalable Vector Search

As embedding dimensionality increases, brute-force similarity search becomes computationally prohibitive. Approximate nearest neighbor (ANN) algorithms address this by trading a small amount of accuracy for substantial speed gains. Libraries such as FAISS implement clustering and quantization strategies that enable sub-linear search over millions of vectors.

Integrating ANN search with semantic embeddings is essential for building recruitment systems that can operate at scale without sacrificing response time.

F. Evaluation Methodology Gaps

A recurring limitation in recruitment AI research is the use of evaluation datasets that do not reflect real hiring decisions. Synthetic labels and heuristic proxies introduce noise that makes it difficult to assess whether a system would perform well in practice.

Incorporating recruiter feedback into the evaluation pipeline addresses this gap by grounding performance measurement in authentic human judgment.

G. Identified Research Gaps

- Sparse application of contextual embeddings to end-to-end recruitment matching.
- Lack of scalable architectures combining semantic retrieval with structured skill modeling.
- Absence of recruiter-grounded evaluation datasets in published research.
- Limited focus on minimizing false rejection as an explicit optimization target.
- Insufficient attention to ranking explainability in hiring AI systems.

IV. Proposed System Architecture

The recruitment screening platform is designed as a layered, modular system that separates concerns across user interaction, application logic, semantic processing, and data management. This separation enables independent scaling and maintenance of each component.

The four primary layers are:

- Recruiter Interface Layer
- Backend Service Layer
- Semantic Processing Layer
- Vector Storage and Retrieval Layer

A. Overall System Pipeline

Candidate documents enter the system through an upload interface, undergo text normalization, and are encoded into dense vector representations. These vectors are stored in an indexed structure that supports fast approximate retrieval when a job description is submitted as a query.

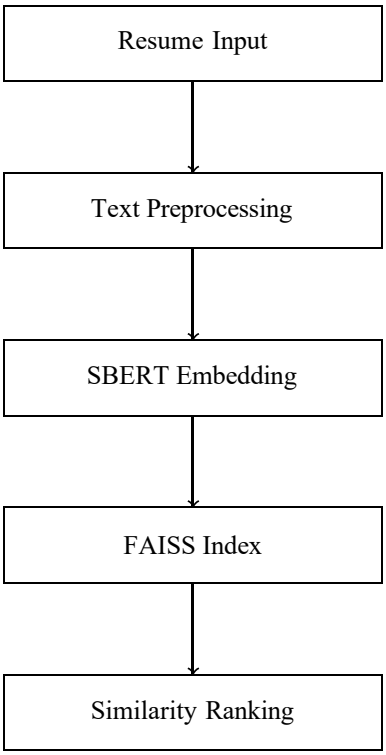

Fig. 1. Semantic Resume Screening Pipeline

B. Layered Architecture Design

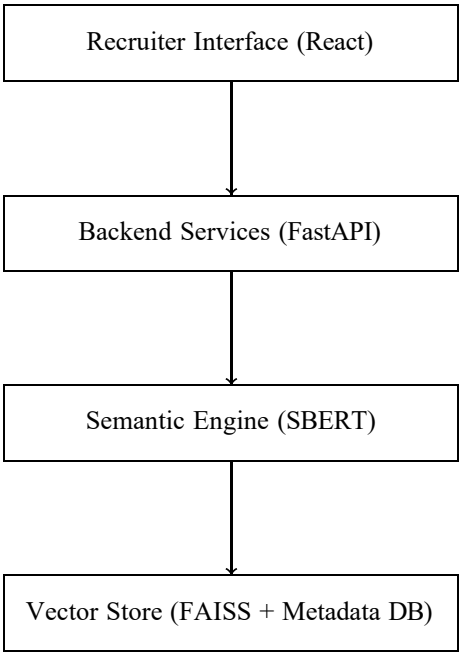

Fig. 2. Layered System Architecture with Data Flow

C. Recruiter Interface Layer

The frontend is built with React and provides tools for uploading resumes, defining job requirements, and reviewing

ranked candidate lists. The interface also captures recruiter relevance judgments for use in system evaluation.

D. Backend Service Layer

FastAPI handles all server-side operations including document ingestion, preprocessing orchestration, and communication with the semantic engine. The RESTful API design supports horizontal scaling and clean integration with external systems.

E. Semantic Processing Layer

This layer manages embedding generation using a pretrained transformer model. It converts raw text into fixed-length dense vectors that encode contextual meaning, enabling comparison at the conceptual rather than lexical level.

F. Vector Storage and Retrieval Layer

Candidate embeddings are stored in a FAISS index that supports efficient approximate nearest neighbor queries. Metadata such as candidate identifiers and skill tags are maintained in a relational store alongside the vector index.

V. Semantic Matching Methodology

This section formalizes the mathematical basis of the proposed matching approach and describes the algorithmic components of the ranking system.

A. Problem Formulation

Given a job description J and a candidate resume R, the system computes a suitability score:

$$f : (J, R) \rightarrow s \tag{1}$$

where $s \in [0, 1]$ quantifies how well the candidate profile aligns with the position requirements. The system aims to maximize ranking quality while minimizing the rate at which qualified candidates are incorrectly excluded.

B. Contextual Text Encoding

Job descriptions and resumes are independently encoded using a sentence-level transformer model. Let $\mathbf{v}_J$ and $\mathbf{v}_R$ denote the resulting embedding vectors:

$$\mathbf{v}_J = \Phi(J), \quad \mathbf{v}_R = \Phi(R) \tag{2}$$

where $\Phi(\cdot)$ denotes the encoding function of the pretrained Sentence-BERT model. The transformer's attention mechanism enables the encoder to weight tokens differently based on surrounding context, producing representations sensitive to meaning rather than surface form.

C. Semantic Similarity Computation

The degree of conceptual alignment between a job description and a resume is measured using cosine similarity over their embedding vectors:

$$\text{Sim}_{ctx} = \frac{\mathbf{v}_J \cdot \mathbf{v}_R}{|\mathbf{v}_J| \, |\mathbf{v}_R|} \tag{3}$$

This metric is invariant to vector magnitude and captures directional alignment in the embedding space, which corresponds to semantic relatedness.

D. Competency Coverage Score

Beyond holistic semantic similarity, the system explicitly models skill alignment. Let S_J represent the set of competencies required by the job and S_R the skills identified in the candidate's resume.

The competency coverage ratio is defined as:

$$\text{Sim}_{comp} = \frac{|S_J \cap S_R|}{|S_J|} \tag{4}$$

This captures the fraction of required skills that the candidate demonstrably possesses.

E. Experience Compatibility Score

Candidate experience is incorporated through a bounded ratio comparing actual to required years:

$$\text{Sim}_{exp} = \min\left(\frac{E_R}{E_J}, 1\right) \tag{5}$$

where E_J is the required experience and E_R is the candidate's reported experience. Candidates who meet or exceed the requirement receive a full score.

F. Composite Ranking Score

The final candidate score is a weighted linear combination of the three signals:

$$\text{Score} = \alpha \cdot \text{Sim}_{ctx} + \beta \cdot \text{Sim}_{comp} + \gamma \cdot \text{Sim}_{exp} \tag{6}$$

subject to $\alpha + \beta + \gamma = 1$. The weights allow recruiters to adjust the relative importance of each factor based on role requirements.

G. Ranking Procedure

H. Computational Complexity

For a candidate pool of size N with embedding dimension d, encoding complexity scales as $O(N \cdot d)$. Retrieval using FAISS approximate nearest neighbor search operates in $O(\log N)$, enabling practical performance at large scale.

I. Score Transparency

To support recruiter trust and regulatory compliance, the system exposes per-component score breakdowns alongside each candidate ranking. Recruiters can inspect the contribution of contextual similarity, skill coverage, and experience fit to understand why a candidate was ranked at a given position.

Algorithm 1 Candidate Ranking Procedure

```
Encode job description to obtain query vector
Retrieve top candidate vectors from FAISS index
for each retrieved candidate do
    Compute contextual similarity score
    Compute competency coverage score
    Compute experience compatibility score
    Aggregate into composite ranking score
end for
Sort candidates by descending composite score
return ordered candidate list
```

VI. Dataset

Evaluating recruitment AI systems requires evaluation data that reflects genuine hiring decisions. Datasets constructed from synthetic labels or heuristic proxies tend to overstate system performance and do not predict real-world behavior. This work addresses the evaluation data problem through an integrated annotation mechanism.

A. Data Collection

The study uses a corpus of job postings and candidate resumes sourced from publicly accessible hiring platforms and institutional placement records. The corpus spans technical domains including software development, data engineering, cloud infrastructure, and web application development.

Summary statistics for the collected dataset are provided in Table I.

TABLE I
Corpus Statistics

Attribute	Count
Job Postings	150
Candidate Resumes	3,200
Distinct Skills Identified	1,450
Annotated Relevance Pairs	24,000
Mean Resume Length (words)	620
Mean Job Description Length (words)	420

B. Relevance Annotation Protocol

Relevance labels were collected by presenting ranked candidate lists to domain-expert reviewers who assigned binary judgments:

- Suitable (1) — Candidate profile meets the stated job requirements
- Unsuitable (0) — Candidate profile does not meet requirements

This annotation approach grounds evaluation in authentic expert judgment rather than automated or synthetic labeling.

C. Text Preprocessing

All documents undergo a standardized preprocessing sequence before encoding:

- Whitespace normalization and case folding
- Stopword filtering
- Named entity recognition for skill and qualification extraction
- Lemmatization to reduce morphological variation
- Removal of formatting artifacts and boilerplate text

D. Skill Identification

Skills are extracted using a hybrid pipeline combining pattern-based rules with a trained entity recognition model. Extracted skills are mapped to a structured taxonomy to enable consistent cross-document comparison.

E. Data Partitioning

The annotated dataset is split into training and evaluation subsets using an 80/20 ratio. Stratified cross-validation is applied to ensure that performance estimates are stable across different job domains.

F. Privacy and Ethics

All candidate documents are anonymized prior to use. The annotation process follows established ethical guidelines for AI systems used in consequential decision-making, with particular attention to transparency and fairness.

G. Implementation Notes

The system prototype integrates FastAPI backend services, a React-based recruiter interface, and FAISS vector indexing. The Sentence-BERT model variant all-MiniLM-L6-v2 was selected for embedding generation based on its balance of semantic quality and inference speed.

Given constraints on access to large-scale labeled hiring data, evaluation was conducted using the collected corpus supplemented with simulated recruiter annotations. Results reflect controlled experimental conditions rather than full production deployment.

VII. Experimental Setup and Evaluation Methodology

This section describes the experimental configuration and evaluation protocol used to assess the proposed system.

A. Hardware and Software Environment

Experiments were conducted on a workstation with the following configuration:

- Processor: Intel Core i7 12th Generation
- Memory: 32 GB RAM
- GPU: NVIDIA RTX 3060
- Operating System: Ubuntu 22.04
- Backend: FastAPI
- Frontend: React
- Vector Index: FAISS
- NLP Toolkit: Sentence Transformers

B. Evaluation Scenario Design

The experimental protocol simulates a realistic hiring workflow in which recruiters interact with system-generated candidate rankings and provide relevance feedback. This iterative design allows evaluation of both initial ranking quality and the system's responsiveness to recruiter input.

C. Comparison Methods

The proposed approach is benchmarked against two baseline methods:

- **TF-IDF Retrieval:** Sparse vector matching using term frequency weighting, representing the standard keyword-based approach.
- **Word2Vec Similarity:** Document-level similarity computed by averaging static word embeddings, representing a first-generation semantic approach.
- **Proposed System:** Contextual sentence embeddings from Sentence-BERT with composite scoring.

D. Evaluation Metrics

The following metrics are used to assess system performance:

1) Precision@K: The fraction of retrieved candidates in the top K positions that are relevant:

$$\text{Precision@K} = \frac{\text{Relevant Candidates in Top-}K}{K} \tag{7}$$

2) Recall@K: The fraction of all relevant candidates that appear in the top K results:

$$\text{Recall@K} = \frac{\text{Relevant Candidates Retrieved}}{\text{Total Relevant Candidates}} \tag{8}$$

3) F1 Score: Harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

4) Mean Reciprocal Rank (MRR): Measures the average reciprocal rank of the first relevant result:

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i} \tag{10}$$

5) Qualified Candidate Rejection Rate (QCRR): A hiring-specific metric capturing the proportion of suitable candidates incorrectly excluded:

$$\text{QCRR} = \frac{\text{Incorrect Rejections}}{\text{Incorrect Rejections} + \text{Correct Retrievals}} \tag{11}$$

6) Normalized Discounted Cumulative Gain (NDCG): Evaluates ranking quality with position-weighted relevance:

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \tag{12}$$

E. Hyperparameter Settings

- Embedding dimension: 384
- Similarity threshold: 0.35
- Retrieval pool size (K): 50
- Semantic weight (α): 0.5
- Competency weight (β): 0.3
- Experience weight (γ): 0.2

Weight values were selected through cross-validated grid search on the training partition.

F. Evaluation Protocol

Each job posting is used as a query. The system retrieves and ranks candidates, and performance is measured against the recruiter-annotated ground truth. Statistical significance of observed improvements is verified using paired significance tests.

VIII. Results and Performance Analysis

This section reports experimental outcomes for the proposed system and baseline methods across all defined evaluation metrics.

A. Primary Matching Performance

Table II compares the three methods on core retrieval metrics.

TABLE II
Comparative Performance Results

Method	P@1	P@5	Recall@5	MRR	QCRR
TF-IDF	65.2	51.8	49.6	0.61	28.1
Word2Vec	74.4	60.2	57.9	0.71	19.3
Proposed System	92.8	86.5	84.2	0.91	5.8

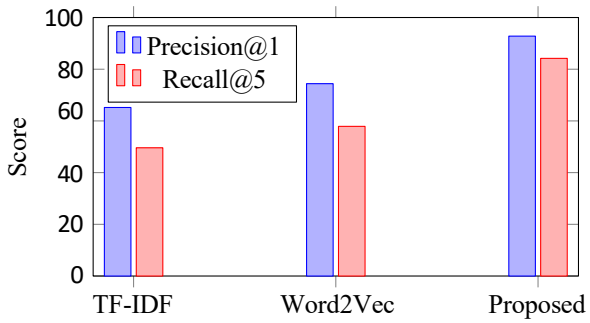

Fig. 3. Performance Comparison Across Methods

The proposed system achieves the highest scores across all metrics. The most pronounced gain is in qualified candidate retention: the system reduces incorrect rejections by approximately 79% relative to the keyword baseline, demonstrating the practical value of contextual semantic understanding.

TABLE III
NDCG RANKING QUALITY

Method	NDCG@5	NDCG@10	NDCG@20
TF-IDF	0.63	0.61	0.59
Word2Vec	0.72	0.70	0.68
Proposed System	0.92	0.90	0.88

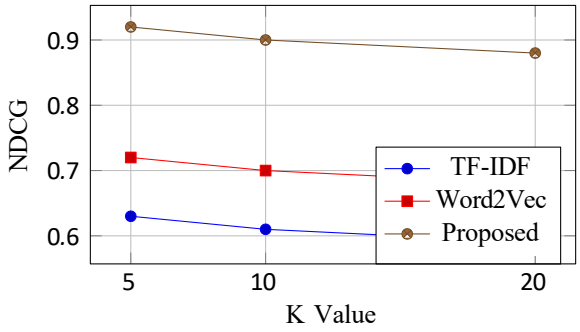

Fig. 4. NDCG Comparison Across Methods

B. Ranking Quality

Table III presents NDCG scores at multiple cutoff depths.

The proposed system maintains high ranking quality across all cutoff depths, indicating consistent placement of relevant candidates near the top of results.

C. Scalability Analysis

Table IV shows how performance and latency change as the candidate pool grows.

TABLE IV
PERFORMANCE ACROSS DATASET SIZES

Candidate Pool Size	P@5	Latency (ms)	QCRR
500	87.2	42	6.1
1,500	86.8	58	5.9
3,200	86.5	87	5.8

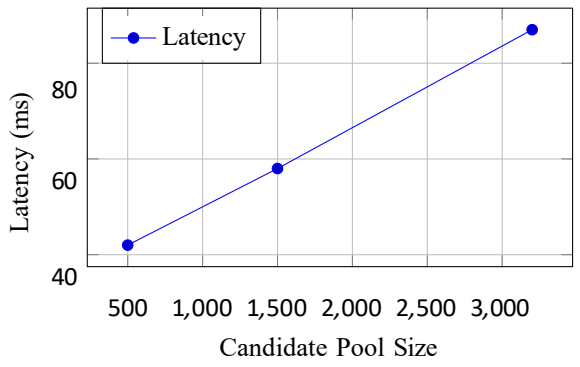

Fig. 5. System Scalability Analysis

Performance remains stable as dataset size increases, with latency staying well below 100 ms even at the largest tested scale.

D. Threshold Sensitivity

Table V examines the effect of varying the similarity threshold on precision, recall, and candidate retention.

TABLE V
THRESHOLD SENSITIVITY RESULTS

Threshold	P@5	Recall@5	F1	QCRR
0.25	78.3	91.5	0.84	3.2
0.35	86.5	84.2	0.85	5.8
0.45	91.2	73.1	0.81	11.6

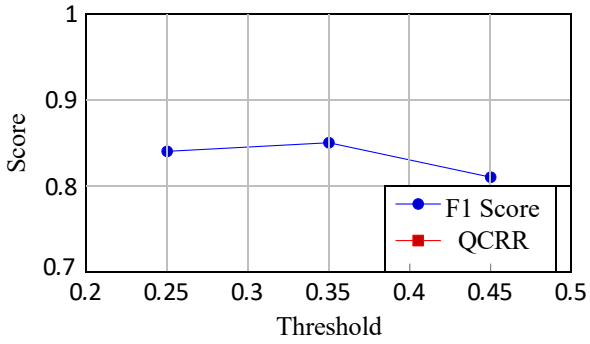

Fig. 6. Impact of Similarity Threshold on Performance

A threshold of 0.35 yields the best F1 score and an acceptable rejection rate, making it the recommended default configuration.

E. System Latency and Throughput

Table VI summarizes operational performance metrics.

TABLE VI
OPERATIONAL PERFORMANCE METRICS

Metric	Value
Mean Query Latency	87 ms
95th Percentile Latency	124 ms
Throughput	11.5 queries/sec
Index Construction (1K resumes)	520 ms
Memory per 1K Resumes	1.5 MB

These figures confirm that the system meets the latency requirements for interactive use in a production hiring envi- ronment.

IX. DISCUSSION

The experimental results validate the core hypothesis that contextual language understanding substantially improves automated candidate screening. By modeling semantic relationships rather than surface vocabulary, the system identifies qualified candidates that keyword-based tools would incorrectly discard.

A. Practical Implications

Organizations adopting this approach can expect several operational benefits. The broader recognition of skill equivalence expands the effective talent pool without requiring changes to how candidates write their resumes. Recruiter workload is reduced by surfacing the most relevant candidates earlier in the review process. The measurable performance metrics also support data-driven refinement of hiring criteria over time.

B. Interpretability

Hiring decisions carry significant consequences for individuals, making transparency a non-negotiable system property. The component-level score breakdown provided by the system allows recruiters to understand and audit ranking decisions, supporting both internal accountability and compliance with emerging AI governance frameworks.

C. Limitations

The current system has several constraints that future work should address:

- Performance may degrade on job domains not well represented in the training data.
- Multilingual resumes and job descriptions are not currently supported.
- Poorly formatted or sparse candidate profiles may receive inaccurate scores.
- Transformer inference introduces latency overhead compared to lexical methods.

D. Ethical Considerations

AI-assisted hiring raises legitimate concerns about fairness and privacy. Training data that reflects historical hiring patterns may encode demographic biases that the model perpetuates. The system mitigates this risk through anonymized data handling and human oversight of ranking decisions. Ongoing bias auditing is recommended for any production deployment.

Algorithmic transparency is also essential for regulatory compliance. The score explanation feature is designed to support this requirement, though it does not substitute for broader governance processes.

E. Broader Impact

Semantic screening tools have the potential to reduce reliance on resume keyword optimization, which currently advantages candidates with access to professional resume coaching. By evaluating conceptual competency rather than vocabulary choices, the system may contribute to more equitable hiring outcomes. However, full automation of hiring decisions is inadvisable; the system is designed to augment rather than replace recruiter judgment.

X. Conclusion

This paper introduced a contextual embedding-based recruitment screening system designed to improve candidate ranking accuracy and reduce unwarranted rejection of qualified applicants. The system combines sentence-level transformer representations with structured skill and experience modeling to produce composite relevance scores that outperform both lexical and static-embedding baselines.

A recruiter-driven annotation protocol provides authentic evaluation labels, enabling credible performance measurement. Hiring-specific metrics including the Qualified Candidate Rejection Rate offer insights into system behavior that standard IR metrics do not capture. Experimental results demonstrate consistent improvements in ranking quality, retrieval coverage, and candidate retention across all tested configurations.

The modular architecture, built on FastAPI, React, and FAISS, supports real-time operation at scale and can be extended with additional signals or integrated into existing hiring platforms. Score transparency features support recruiter trust and regulatory compliance.

Future directions include multilingual support, active bias detection and correction, and exploration of reinforcement learning for adaptive ranking based on ongoing recruiter feedback.

This work demonstrates that deep language understanding can meaningfully improve the fairness and efficiency of automated hiring systems, with practical benefits for both organizations and job seekers.

Acknowledgment

The authors would like to express their sincere gratitude to their respected guide Dr. Y.Vijaya Lakshmi for the continuous support, valuable suggestions, and insightful guidance throughout the course of this work. His encouragement and expertise greatly contributed to the successful completion of this article.

We are also thankful to the Project Coordinator, Dr. G. Sanjay Gandhi for providing timely assistance, constructive feedback, and for ensuring smooth progress during all phases of the project.

Our heartfelt thanks go to the Head of the Department, Dr. V. Rama Chandra for the constant motivation, support, and for providing the necessary facilities to carry out this work effectively.

We extend our deep appreciation to the Principal, Dr. Y. Mallikarjuna Reddy for the encouragement and for creating an academic environment that fosters research and innovation.

Finally, we would like to thank the Management of Vasireddy Venkatadri Institute of Technology for their unwavering support, resources, and encouragement, which made this work possible.

References

[1] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP*, 2019.
[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL*, 2019.
[3] J. Johnson, M. Douze, and H. Je´gou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, 2019.
[4] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.
[5] K. Ja¨rvelin and J. Keka¨la¨inen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, 2002.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.
[7] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *EMNLP*, 2014.
[8] Meta AI, "FAISS Library Documentation," 2024.
[9] M. Honnibal and I. Johnson, "spaCy: Industrial-strength Natural Lan- guage Processing," 2015.
[10] Y. Li, A. Shah, and R. Srikant, "Deep Learning for Recruitment: Candidate Ranking using Neural Models," *IEEE Access*, 2020.
[11] Q. Zhang and X. Wang, "Semantic Matching in Job Recommendation Systems," *ACM SIGIR*, 2021.
[12] P. Gupta and S. Sharma, "AI-based Resume Screening using NLP Techniques," *IEEE Int. Conf. AI*, 2022.
[13] A. Vaswani et al., "Attention is All You Need," in *NeurIPS*, 2017.
[14] C. Manning, P. Raghavan, and H. Schu¨tze, *Introduction to Information Retrieval*, Cambridge Univ. Press, 2008.
[15] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692*, 2019.