Resume Screening System with Semantic Search, Bias-Mitigation, and Explainability

Ms. Priya Chanda[1], T Bhavya Sri[2], C Tejasree[3], N Mourya Vardhan[4], M Rohith[5]

Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be) University, Bangalore, India

Abstract - The modern recruitment landscape is characterized by an unprecedented volume of digital applications, often overwhelming human resource departments. While Applicant Tracking Systems (ATS) have been widely adopted to automate the initial screening process, legacy systems utilizing keyword-based filtering (Boolean logic and TF-IDF) suffer from critical limitations. These systems frequently reject qualified candidates due to vocabulary mismatches and fail to detect systemic biases embedded within job descriptions. This research presents the design and implementation of a comprehensive, AI-driven Resume Screening System that transcends simple keyword matching. The proposed framework integrates **Natural Language Processing (NLP)**, **Transformer-based Sentence Embeddings (Sentence-BERT)**, and a novel **Bias Audit Module**. The system employs a three-tier architecture: (1) A Semantic Matching Engine that maps resumes and job descriptions to a high-dimensional vector space to capture contextual meaning; (2) A Bias Audit layer that proactively detects and neutralizes gender-coded language in job descriptions; and (3) An Explainable AI (XAI) interface that visualizes scoring logic for recruiters. Experimental results on a dataset of 2,400+ resumes demonstrate a **26% improvement in retrieval precision** over traditional baselines and confirm the system's capability to mitigate linguistic bias, thereby promoting a fairer, more transparent, and efficient hiring ecosystem.

Keywords—Natural Language Processing (NLP), Sentence-BERT, Semantic Similarity, Explainable AI (XAI), Bias Mitigation, Applicant Tracking Systems (ATS), Named Entity Recognition (NER), Deep Learning.

I. Introduction

The digitization of the recruitment industry has democratized job access, allowing candidates to apply for positions globally with minimal friction. However, this ease of access has created a "data deluge" for recruiters. Industry reports indicate that a standard corporate job opening attracts an average of 250 resumes. Of these, only 4 to 6 candidates are typically interviewed, and only one is hired. The manual effort required to filter the initial pool is substantial, often estimated at up to 23 hours for a single hire. This repetitive task is not only labor-intensive but also prone to human fatigue and cognitive biases, leading to inconsistent evaluation standards.

To manage this volume, organizations rely on Applicant Tracking Systems (ATS). However, the majority of commercially available ATS solutions are built on archaic Information Retrieval (IR) principles.

Lexical Rigidity (The Vocabulary Mismatch Problem): Traditional systems use Boolean search logic or Term Frequency-Inverse Document Frequency (TF-IDF) to rank candidates. These algorithms rely on exact string matching. Consequently, a candidate describing their experience as "building predictive models" might be rejected if the Job Description (JD) strictly filters for the keyword "Machine Learning," despite the concepts being semantically identical. This results in a high rate of False Negatives (qualified candidates being rejected).
Bias Propagation: Most screening algorithms ingest Job Descriptions without scrutiny. Research has shown that JDs often contain "gender-coded" language (e.g., "ninja," "dominant," "aggressive") that unconsciously deters female applicants. Standard ATS tools propagate this bias by ranking candidates who mirror this aggressive language higher, thereby reinforcing the lack of diversity in technical fields.
The "Black Box" Problem: Deep learning models, while accurate, often lack transparency. Recruiters are presented with a ranked list without understanding *why* a specific candidate is ranked #1. This lack of explainability erodes trust in automated systems and raises compliance issues under regulations like GDPR.

Research Objectives

This study aims to develop a holistic screening framework that moves beyond keywords to "concept matching." The specific objectives are:

To implement **Named Entity Recognition (NER)** for structured information extraction (Skills, Education, Experience) from unstructured resume text.

To utilize **Sentence-BERT (SBERT)** to generate dense vector representations of resumes, facilitating semantic similarity calculation.

To integrate a **Bias Audit Layer** that evaluates Job Descriptions for exclusionist terminology before the screening process begins.

To provide **Explainability** through visual highlighting of matched terms and score decomposition, ensuring recruiters understand the AI's decision-making process.

II. Literature Survey

The evolution of resume screening technology can be categorized into three generations: Keyword-based, Statistical Machine Learning, and Deep Learning. **Table 1** summarizes key contributions and identified gaps.

Table 1: Comparative Analysis of Existing Literature

Author(s)	Methodology	Key Contributions	Limitations / Gaps Identified
Gupta & Verma (2022) [1]	Automated Parsing using TF-IDF and Boolean Logic.	Established baseline efficiency for resume parsing; effective for exact keyword retrieval.	Fails to capture context; ignores word order; high rejection rate for synonyms.
Sinha et al. (2023) [3]	Domain Prediction using SVM and Random Forest classifiers.	Successfully categorized resumes into broad domains (e.g., IT, Sales) with high accuracy.	Requires extensive feature engineering; struggles with layout variance in PDFs; does not perform fine-grained ranking.
Deshmukh & Raut (2024) [5]	BERT-based Candidate Ranking.	Utilized standard BERT embeddings to capture context, outperforming static embeddings like GloVe.	Standard BERT is computationally expensive for pair-wise comparison of large datasets (Cross-Encoder bottleneck).
Reimers & Gurevych (2019) [2]	Sentence-BERT (SBERT).	Introduced Siamese Networks to optimize BERT for cosine similarity tasks, reducing inference time.	While a seminal paper on SBERT, it was not applied specifically to the domain of recruitment bias mitigation.
Mehrabi et al. (2021) [7]	Survey on Bias in ML.	Categorized types of bias (historical, representation, aggregation) in AI systems.	Theoretical survey; did not propose a specific system architecture to mitigate bias in active ATS pipelines.
Sriranjani & Kalaiarasi (2025) [4]	Ethical Concerns in AI Screening.	Highlighted the risks of "Black Box" algorithms in hiring and GDPR compliance.	Focused on the ethical framework but lacked a technical implementation of an Explainable AI (XAI) solution.

Summary of Research Gap: While recent works have applied BERT to resume screening, there is a conspicuous lack of unified systems that combine *semantic accuracy* (SBERT) with *active bias mitigation* and *explainability*. Most systems focus purely on ranking accuracy, ignoring the ethical implications of the input data (Job Descriptions).

III. Theoretical Framework

Deep Learning for NLP: From Word2Vec to Transformers

Early NLP relied on "Bag of Words" models, which discarded word order and context. Word2Vec introduced static embeddings, where words were mapped to vectors, but these were non-contextual (the word "bank" had the same vector in "river bank" and "bank deposit"). The **Transformer** architecture [10] revolutionized this by using "Self-Attention" mechanisms, allowing the model to weigh the importance of different words in a sentence relative to each other, providing dynamic, contextual embeddings.

Sentence-BERT (SBERT) Architecture

Standard BERT (Bidirectional Encoder Representations from Transformers) is powerful but inefficient for semantic search. To compare a JD against 1,000 resumes, a standard BERT model would need to pair the JD with every resume and pass them through the massive network layers individually.

We utilize **SBERT** [2], which modifies the BERT architecture using **Siamese Networks**.

Siamese Structure: The network consists of two identical BERT towers with shared weights.
Input Processing: Sentence A (Job Description) is processed by Tower 1, and Sentence B (Resume) by Tower 2.
Pooling Layer: The output of the BERT layers is "pooled" (Mean Pooling) to create a fixed-size sentence embedding vector (*u* and *v)*
Cosine Similarity: The semantic similarity is calculated using the cosine angle between vectors *u* and *v*:

$$\text{Similarity}(u, v) = \frac{u \cdot v}{\|u\|\|v\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

This architecture allows us to pre-compute embeddings for all resumes, reducing the search process to a fast mathematical calculation in vector space.

Bias in Language Models

Research by Gaucher et al. [9] demonstrates that specific words in job advertisements serve as "institutional gatekeepers." Words like *competitive, dominant, leader* are socially coded as masculine, while *support, community, understand* are coded as communal. An AI system trained on historical data often learns to associate technical roles with agenticterms. Our framework addresses this theoretically by auditing the *input query* (the JD) before it interacts with the vector space, thereby sanitizing the signal at the source.

IV. Proposed Methodology

The system architecture is designed as a modular pipeline consisting of five distinct stages.

Step 1: Data Ingestion & Text Extraction

Resumes are unstructured data usually in PDF or DOCX format.

PDF Parsing: We utilize the *pdfminer.six* library to extract text. A custom coordinate-based heuristic is applied to detect multi-column layouts, which are common in modern resumes. This prevents the "lines reading across columns" error that often confuses standard parsers.
Normalization: Extracted text is normalized (lowercased), and noise (bullets, non-ASCII characters) is removed. Crucially, technical punctuation (e.g., C++, C#, .NET) is preserved to maintain skill distinctiveness.

Step 2: Named Entity Recognition (NER)

To extract structured data for the user interface, we employ a **spaCy** pipeline customized for the HR domain.

Training Data: The model is fine-tuned on an annotated dataset where entities are labeled as SKILL, DEGREE, ORG, DESIGNATION, and EXPERIENCE.
Hybrid Approach: In addition to statistical NER, we use Rule-Based Matchers for skills that have high variance in spelling (e.g., "ReactJS", "React.js", "React") to ensure high recall rates.

Step 3: The Bias Audit Module

Before the screening initiates, the system analyzes the Job Description (JD).

Tokenization: The JD is broken into tokens.

Lexicon Matching: Tokens are compared against a curated database of gender-coded terminology derived from social science research.

Scoring Metric:

M_{count} = Count of Agenticwords

F_{count} = Count of Communal words

If (M_{count} - F_{count}) > Threshold, the system flags the JD as "Biased."

Mitigation: The system suggests neutral synonyms (e.g., suggesting "collaborative" instead of "aggressive" or "track record" instead of "proven dominance").

Step 4: Semantic Vectorization and Ranking

This is the core retrieval engine.

Model Selection: We utilize the *all-MiniLM-L6-v2* model from HuggingFace. This model is a distilled version of BERT, offering high speed (inference < 100ms) with minimal loss in accuracy compared to larger models.

Vector Space Mapping: The Job Description is converted into a vector V_{jd} (384 dimensions). Every extracted resume text is similarly converted into a vector V_{res}.

Ranking Algorithm: Semantic Ranking Logic

Input: Job_Desc_Text, List_of_Resumes
Output: Ranked_List
1. clean_JD = Preprocess(Job_Desc_Text)
2. vector_JD = SBERT_Encode(clean_JD)
3. scores = []
4. For each resume in List_of_Resumes:
   a. clean_res = Preprocess(resume.text)
   b. vector_res = SBERT_Encode(clean_res)
   c. sim_score = Cosine_Similarity(vector_JD, vector_res)
   d. scores.append(sim_score)
5. Sort List_of_Resumes by scores descending
6. Return Top_K Results

Step 5: XAI-Recruiter (Explainability)

To build trust, the system provides interpretability for the generated scores.

Keyword Highlighting: We use the **YAKE!** (Yet Another Keyword Extractor) algorithm to identify the most salient terms in the JD and verify their presence in the resume, highlighting matches in green and missing key terms in red.

Visualization: A radar chart compares the candidate's skills against the JD's required skill clusters, providing a visual gap analysis for the recruiter.

V. Experimental Results and Discussion

Dataset Description

The system was evaluated using the **Kaggle Resume Dataset**, which contains 2,484 resumes labeled across 24 distinct categories (e.g., Data Science, HR, Arts, Mechanical Engineering). To rigorously test the Bias Audit module, we also synthesized 50 Job Descriptions with varying levels of linguistic bias and technical requirements.

Performance Metrics

We evaluated the retrieval performance using **Precision@K** (P@K) and **Normalized Discounted Cumulative Gain (NDCG)**.

Precision@K: Measures the proportion of relevant candidates found in the top K results.

NDCG: Measures the quality of the ranking order (i.e., are the *best* candidates at the very top?).

Table 2: Performance Comparison of Retrieval Models

Metric	TF-IDF (Baseline)	Word2Vec (Avg)	Proposed System (SBERT)
Precision@5	0.58	0.65	0.84
Precision@10	0.52	0.61	0.79
NDCG	0.60	0.68	0.82
Inference Time (100 docs)	0.05s	0.8s	2.1s

Analysis: The Proposed System significantly outperforms the TF-IDF baseline. TF-IDF struggled heavily with resumes that had diverse vocabulary. For example, a resume mentioning "Statistical Analysis" and "Regression" was missed by TF-IDF for a "Data Scientist" role (which searched for those exact terms) but was correctly ranked by SBERT due to the proximity of these vectors in the embedding space. While the inference time is slightly higher for SBERT, it remains well within acceptable limits for real-time applications.

Bias Mitigation Analysis

We ran the Bias Audit Module on a set of 20 real-world "Software Engineering" job descriptions scraped from the web.

Initial State: 14 out of 20 JDs were flagged for heavy agenticcoding (Average score: 75% Masculine). Common terms included "ninja," "rockstar," and "dominate."
Intervention: The system suggested neutral replacements (e.g., "Developer" instead of "Ninja," "Lead" instead of "Dominate").
Outcome: After neutralizing the JDs, we simulated a candidate pool application using a diverse set of synthetic resumes. The neutralized JDs attracted a more balanced distribution of semantic matches, proving that removing gendered language reduces the "filtering out" of diverse candidates at the semantic level.

Computational Efficiency

The system was tested on a local server with a standard consumer-grade GPU (NVIDIA GTX 1660 Ti).

PDF Parsing Time: Average 0.5 seconds per file.
Vector Inference Time: Average 0.02 seconds per file.
Total Batch Processing: Processing a batch of 100 resumes takes approximately 55 seconds. This demonstrates that the system is scalable for real-time enterprise use cases without requiring expensive high-performance computing clusters.

VI. Conclusion and Future Scope

This research paper presented the design, implementation, and validation of an AI-driven Resume Screening System. By integrating **Sentence-BERT**, we successfully addressed the limitations of keyword-based ATS, achieving a **26% increase in Precision@5**. More importantly, the integration of the **Bias Audit Module** and **XAI visualizations** addresses the critical ethical concerns regarding fairness and transparency in automated hiring. The system acts not as a replacement for human recruiters, but as an intelligent, fair, and explainable assistant that handles the high volume of data, allowing humans to focus on the qualitative and interpersonal aspects of hiring.

Generative AI Integration: Future iterations will explore replacing the ranking score with a generative text summary using Large Language Models (e.g., GPT-4 or Llama-3), providing recruiters with a written paragraph explaining *why* a candidate is a good fit.
Multimodal Screening: Extending the system to analyze candidate portfolios (GitHub links, Behance portfolios) and video introductions using multimodal neural networks to capture soft skills.

Blind Screening Automation: Automatically anonymizing resumes (removing names, gender, college names, and locations) before the ranking phase to further ensure zero-bias evaluation during the initial screening pass.

References

[1] A. Gupta and R. Verma, "Automated Resume Parsing: A Natural Language Processing Approach," *International Journal of Computer Applications*, vol. 183, no. 4, pp. 12-18, 2022.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[3] A. K. Sinha, M. A. K. Akhtar, and M. Kumar, "Automated Resume Parsing and Job Domain Prediction using Machine Learning," *Indian Journal of Science and Technology*, vol. 16, no. 26, pp. 1967-1974, 2023.

[4] S. K. Sriranjani and H. Kalaiarasi, "AI-BASED RESUME SCREENING SYSTEMS: OPPORTUNITIES AND ETHICAL CONCERNS," *International Journal for Research Trends and Innovation*, vol. 10, no. 6, pp. c141-c143, 2025.

[5] A. Deshmukh and A. B. Raut, "Applying BERT-Based NLP for Automated Resume Screening and Candidate Ranking," *Annals of Data Science*, vol. 12, no. 2, pp. 591–603, 2024.

[6] S. Barocas and A. D. Selbst, "Big data's disparate impact," *California Law Review*, vol. 104, p. 671, 2016.

[7] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1-35, 2021.

[8] C. O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group, 2016.

[9] D. Gaucher, J. Friesen, and A. C. Kay, "Evidence that gendered wording in job advertisements exists and sustains gender inequality," *Journal of Personality and Social Psychology*, vol. 101, no. 1, pp. 109–128, 2011.

[10] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[12] F. Campos et al., "YAKE! Keyword extraction from single documents using multiple local features," *Information Sciences*, vol. 509, pp. 257-289, 2020.