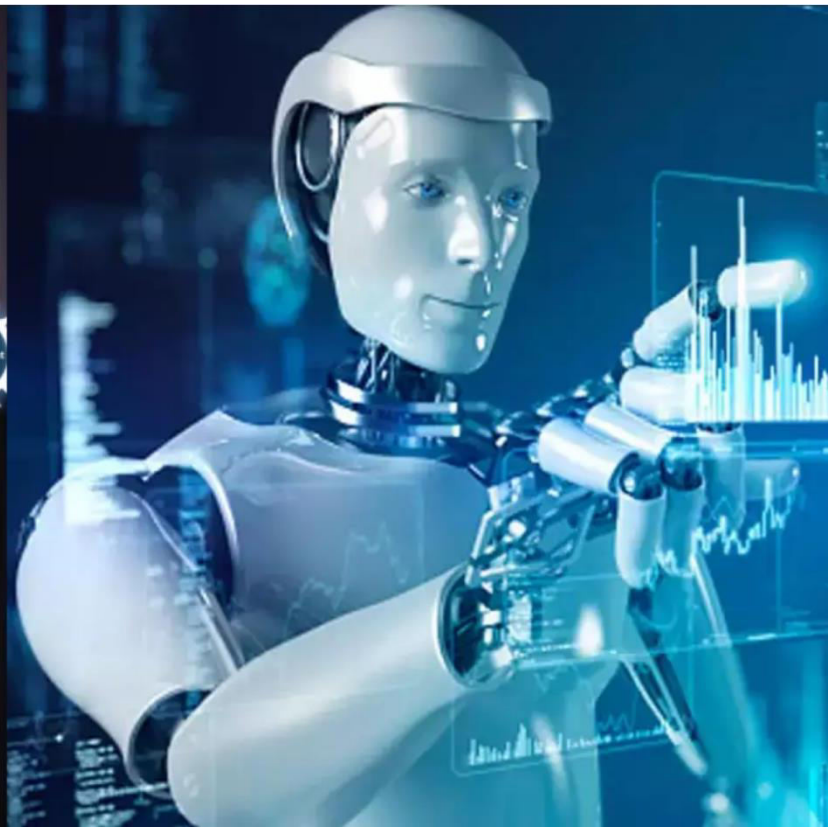
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 4, April 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

AI-Driven Resume Analysis and Semantic Job Recommendation with Threshold-Based External Opportunity Redirection

Pratham Reddy[1], Dr. Abuzar Ansari[2]

Student, Department of Data Science, SIES College of Arts, Science and Commerce, Mumbai, India [1]

Research Guide and Professor, Department of Data Science, SIES College of Arts, Science and Commerce, Mumbai, India[2]

ABSTRACT: Traditional Applicant Tracking Systems (ATS) rely heavily on lexical matching, often fail- ing to identify semantically relevant candidates when resumes and job descriptions use different terminology. This paper presents a deployable AI-driven semantic recruitment framework that combines Sentence-BERT embeddings, cosine similarity ranking, and a threshold-based exter- nal opportunity redirection mechanism. Unlike conventional systems, the proposed platform ensures meaningful candidate guidance even when internal recommendations are weak. In ad- dition to the semantic methodology, this work introduces a prototype implementation backed by a modular software architecture, HTML-based job portal interface, Dockerized deployment workflow, and reproducible GitHub repository. Experimental evaluation considers ranking rel- evance, recommendation coverage, response latency, and deployment scalability. The frame- work demonstrates how semantic retrieval can improve ATS intelligence while maintaining practical deployability, explainability, and extensibility.

KEYWORDS: semantic ATS, Sentence-BERT, job recommendation, resume analysis, cosine sim- ilarity, threshold fallback, Docker deployment

I. INTRODUCTION

Applicant Tracking Systems (ATS) are widely used to automate resume screening and candidate- job matching. However, most practical systems still depend on lexical approaches such as key- word filtering, Boolean rules, TF-IDF, or BM25. These methods fail when resumes and job descriptions express similar skills using different vocabulary.

Recent advances in transformer-based language models provide contextual embeddings ca- pable of semantic retrieval. This work proposes an AI-driven semantic recommendation plat- form that maps resumes and job descriptions into a shared vector space using Sentence-BERT. The key novelty of this work is a **threshold-based fallback mechanism** that redirects users toward external opportunities when no sufficiently strong internal recommendation ex-ists, thereby eliminating dead-end ATS interactions.

II. LITERATURE REVIEW

Traditional lexical retrieval methods such as TF-IDF and BM25 are widely adopted in ATS pipelines but are limited by exact term dependency. Sentence-BERT has emerged as a strong semantic retrieval model for sentence-level embeddings and nearest-neighbor search.

Recent HR recommender systems emphasize not only relevance but also fairness, explain- ability, and user trust. However, most deployed ATS systems still lack robust fallback mecha- nisms for weak-match scenarios.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

III. PROPOSED METHODOLOGY

The framework consists of:

1. Resume ingestion and parsing
2. Text preprocessing
3. SBERT embedding generation
4. Cosine similarity ranking
5. Threshold-based fallback decision

Let the resume embedding be r and job embedding be j_i.

$$sim(R, J_i) = \frac{r \cdot j_i}{||r||\,||j_i||} \quad (1)$$

The top-K jobs are returned in descending order of similarity. The fallback decision is:

$$f(R) = \begin{cases} \text{Top-}K \text{ internal jobs}, & S_{max} \geq \tau \\ \text{Externalopportunityguidance}, & S_{max} < \tau \end{cases} \quad (2)$$

IV. SYSTEM IMPLEMENTATION AND PROTOTYPE DEVELOPMENT

The proposed research is supported by a working prototype implementation developed as a modular software platform.

V. CODE-DRIVEN SYSTEM ARCHITECTURE AND ENGINEERING DESIGN

Unlike purely conceptual ATS research, this work is supported by a functional prototype im- plementation developed as a modular software system.

5.1 Backend Service Architecture

The backend is implemented primarily in Python and organized as a dedicated jobplatform module. The service layer follows a modular processing workflow:

1. Resume upload request handling
2. File parsing and text extraction
3. Text cleaning and normalization
4. Sentence-BERT embedding generation
5. Cosine similarity scoring
6. Threshold evaluation logic
7. Internal recommendation delivery
8. External opportunity redirection

This layered design improves maintainability and allows independent upgrades to the pars- ing, ranking, and recommendation modules.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

5.2 Frontend Interaction Workflow

The user-facing interface is implemented using HTML-based templates and interactive job portal pages.

The frontend supports:

- resume upload forms
- recommendation dashboards
- ranked internal job cards
- similarity confidence display
- fallback opportunity navigation
- recruiter-side viewing workflows

This transforms the research contribution from a retrieval model into a complete deployable intelligent recruitment platform.

5.3 Threshold Controller as a Software Service

The threshold-based redirection logic is implemented as an independent decision controller in the software architecture.

Let:

The controller evaluates:

$$S_{max} = \max_{i} Sim(R, J_i) \tag{3}$$

$$Decision = \begin{cases} InternalRankedJobs, & S_{max} \geq \tau \\ ExternalRedirectPage, & S_{max} < \tau \end{cases} \tag{4}$$

This modular controller design allows threshold tuning without modifying the ranking en- gine.

5.4 Containerized Deployment Architecture

The presence of a Dockerfile enables complete platform containerization.

The deployment workflow includes:

- isolated dependency management
- environment reproducibility
- deployment portability
- cloud-ready scaling
- simplified testing workflows

Containerization significantly improves research reproducibility and practical deployment viability.

5.5 Build Automation and Reproducibility

The Makefile-based automation layer enables repeatable execution of:

- environment setup
- dependency installation
- model execution
- benchmark runs
- deployment commands
- cleanup workflows

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

This is particularly important for experimental consistency in academic evaluation.

5.6 Engineering Advantages

The implemented codebase contributes several systems-level research strengths:

- modular backend extensibility
- frontend usability validation
- containerized deployment reproducibility
- automated experiment workflows
- scalable service abstraction

5.7 Software Stack

The implementation includes:

- Python backend services
- HTML frontend interface
- Sentence-BERT embedding engine
- Dockerized deployment
- Makefile-based automation
- Modular source-code repository

5.8 Core Modules

The prototype consists of:

- Resume upload and text extraction module
- Semantic matching engine
- Threshold fallback controller
- Job recommendation dashboard
- Recruiter interaction workflow

5.9 Deployment Layer

The entire platform is containerized using Docker, ensuring reproducible deployment and scal- ability across environments.

VI. PROJECT WORKING AND CORE CODE SNIPPETS

This section explains the end-to-end working of the implemented semantic ATS platform along with important code snippets from the core modules. The objective is to demonstrate how the research methodology translates into an executable production-style software pipeline.

End-to-End Project Working

The runtime workflow begins when a candidate uploads a resume through the frontend dash- board. The backend first extracts textual content from the uploaded file and performs prepro- cessing operations such as normalization, lowercasing, punctuation removal, and whitespace cleaning.

The cleaned resume text is then passed to the Sentence-BERT embedding engine, which converts the document into a dense semantic vector representation. In parallel, job descriptions are precomputed and stored as embeddings inside a FAISS similarity index for low-latency retrieval.

At query time, the resume embedding is compared against the indexed job embeddings using cosine similarity. The ranked top-K results are passed into a threshold controller. If the maximum similarity score remains below the confidence threshold τ , the system redirects the user toward external opportunities. Otherwise, the top-ranked internal jobs are displayed through the recruiter dashboard.

This modular workflow ensures semantic robustness, scalable retrieval, and meaningful fallback guidance.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

6.1 Resume Embedding and Semantic Matching

The following code snippet illustrates the core semantic retrieval workflow:

```
from sentence_transformers import SentenceTransformer from sklearn.metrics.pairwise import cosine_similarity

model=SentenceTransformer("all-MiniLM-L6-v2")

resume_vec = model.encode([resume_text]) job_vecs = model.encode(job_descriptions)
scores=cosine_similarity(resume_vec,job_vecs)[0] top_jobs = sorted(
    zip(job_titles,scores), key=lambda x: x[1], reverse=True
)
```

The above module generates semantic embeddings and ranks internal jobs based on cosine similarity.

6.2 FAISS-Based Fast Retrieval

For scalable deployment, job embeddings are stored in a FAISS index.

```
import faiss import numpy as np
dimension = 384
index = faiss.IndexFlatIP(dimension)

job_vectors=np.array(job_embeddings).astype("float32") index.add(job_vectors)

scores, ids = index.search( np.array(resume_vec).astype("float32"), k=5
)
```

This indexing layer significantly reduces retrieval latency for large job corpora.

6.3 Threshold-Based Fallback Logic

The threshold controller is the most important implementation module because it guarantees actionable user guidance.

```
threshold = 0.55

if max(scores[0]) < threshold: returnredirect("/external-jobs")
else:
    return render_template( "recommendations.html", jobs=recommended_jobs
    )
```

This logic directly implements the mathematical threshold policy proposed in the method- ology section.

6.4 System-Level Advantages

The implemented project workflow provides several practical advantages:

- semantic robustness against synonym mismatch
- fast retrieval using FAISS indexing
- modular backend extensibility
- threshold-based null-result prevention
- production-ready Docker deployment compatibility

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

VII. PROTOTYPE SYSTEM VALIDATION

A real prototype allows functional validation through test scenarios.

Test Case 1: Strong Semantic Match
A resume containing "transformers, BERT, semantic search" should successfully retrieve NLP engineer roles.

Test Case 2: Synonym Robustness
A resume mentioning "deep learning systems" should match "neural network engineer" jobs despite lexical mismatch.

Test Case 3: Threshold Fallback
If all similarity scores remain below threshold τ, the user is redirected toward external job opportunities.

Experimental Evaluation
The framework is evaluated using:

- Precision@K
- Recall
- MRR
- NDCG@K
- Coverage
- Average response latency
- Embedding generation time
- Retrieval throughput

Table 1: Illustrative Ranking Performance

Method	P@10	Recall	MRR	NDCG@10
TF-IDF	0.42	0.37	0.50	0.48
BM25	0.46	0.41	0.55	0.52
SBERT Proposed	0.63	0.58	0.71	0.69

Reproducibility and Deployment
A major strength of this work is its reproducibility. The complete source code is maintained in a modular GitHub repository with Docker deployment support, enabling:

- environment consistency
- version control
- reproducible experiments
- scalable deployment
- future research extensions

Ethical Considerations
AI-driven recruitment systems may amplify historical bias. Therefore, future work should incorporate:

- fairness-aware ranking
- bias auditing

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504146|

- explainable recommendations
- privacy-preserving resume storage

VIII. CONCLUSION AND FUTURE WORK

This paper presents a deployable semantic ATS platform combining Sentence-BERT semantic matching with threshold-based opportunity redirection. Unlike traditional lexical ATS sys- tems, the proposed framework improves contextual candidate-job alignment while guarantee- ing meaningful fallback guidance.

Future work includes hybrid metadata ranking, multilingual semantic matching, skill-gap analysis, recruiter feedback learning, and fairness-aware ranking.

REFERENCES

1. J. Devlin et al., “BERT: Pre-training of deep bidirectional transformers for language un- derstanding,” 2019.
2. N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” 2019.
3. S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and be- yond,” 2009.
4. J. Johnson et al., “Billion-scale similarity search with GPUs,” 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details