Priyanka Narang, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Beyond Heuristics: A Data-Driven Hybrid Architecture for Semantic Resume Analysis and Context-Aware Explainable AI

Priyanka Narang[1], Tathya Sharma[2], Aditi Kashyap[3], Mamta[4]
[1,2,3]Student, AIML Department, HMRITM, Hamidpur, New Delhi, INDIA
[4]Assistant Professor, Management Department, HMRITM, Hamidpur, New Delhi, INDIA

Abstract- Traditional Applicant Tracking Systems (ATS) and open source resume analyzers rely on rigid regex rules and hardcoded penalty schemes. As a result, they are brittle, and candidates are penalized for using different but equivalent terminology. These systems often fail to reflect how skills and roles actually appear in real industry data. This paper presented a fully data-driven multi-model machine learning pipeline that replaced arbitrary scoring rules with seven coordinated models. A Multi-Stage Waterfall Extraction model combined a custom spaCy named entity recognition system with SBERT-based cosine similarity to overcome sparse and noisy annotations. An XGBoost-driven linear regression severity engine and an implicit ontology inference module revealed invisible ATS filters rooted in real hiring expectations. A deterministic Explainable AI layer then grounded all feedback in the candidate's own verified text, removing operational hallucinations. The paper reported empirical results including the Soft Skill Anomaly, which challenges common developer assumptions about soft skills in ATS design.

Keywords: Semantic Resume Analysis, Named Entity Recognition, Explainable AI.

I. INTRODUCTION

In a modern recruitment pipeline, recruiters should not have to manually review hundreds of applications, as one average corporate role receives around 250 resumes, four to six candidates make it to the interview stage, and only one is hired, which could take up to 23 hours of work per hire [9].

To automate the early stages of this process, Applicant Tracking Systems (ATS) were developed, but most legacy ATS tools still use keyword matching techniques, such as Boolean search and TF-IDF vectorization, which are too rigid to capture the semantic richness of human-written resumes and job descriptions [3].

Three structural problems explain the shortcomings of current systems in practice: semantic rigidity, where binary keyword matching does not account for conceptually equivalent skills (e.g., a candidate who lists Neural Networks may be penalized for not listing Deep Learning, which is functionally equivalent [9, 3]); heuristic severity bias, where penalty scores in legacy systems are hardcoded by developers based on intuition, rather than on analysis of real labor market data, and no published architecture has replaced these fixed rules with a supervised data-driven severity function; and generative AI hallucination, where modern LLM-based feedback tools are susceptible to producing misinformation [5]. This raises clear issues of trust and ethics in a candidate feedback setting, as models reward candidates for claiming skills they do not possess.

This paper presents a machine learning architecture composed of seven models that is designed to address three common failure modes. Rather than relying on heuristic brackets, the system uses data driven components, includes implicit job description inference trained on real world resume frequency data, and features a deterministic XAI layer that ensures all feedback is based on the candidate's own verified text.

II. LITERATURE REVIEW AND RESEARCH GAPS

This section examines four key areas: keyword based ATS and NLP resume parsing, semantic matching through transformer embeddings, implicit skill

inference from job descriptions, and Explainable AI for candidate feedback. Each subsection concludes by identifying the specific research gap that the proposed architecture is intended to address.

Keyword-Based ATS and NLP Resume Parsing

Early resume screening systems relied on Bag of Words models and TF IDF vectorization to evaluate how closely the language in resumes matched the language in job descriptions [2]. Named Entity Recognition later emerged as an important improvement. Sajid et al. [10] proposed a three stage NER framework for skill extraction, while Mohanty et al. [7] combined NER with traditional classifiers such as KNN, SVM, Decision Tree, Naive Bayes, and XGBoost to categorize resumes. Liang et al. [6] introduced a multitask deep learning model that performs resume segmentation and entity recognition simultaneously and reported improved accuracy across both English and Chinese corpora. Despite these advances, a recurring limitation across these approaches is the way they handle low NER F1 scores on sparsely annotated datasets.

These scores are often treated as a fixed performance limit rather than as evidence that the underlying architecture may need to be restructured to address sparsity more effectively. Existing NER based resume parsers frequently treat weak performance on sparsely annotated HR datasets as an unavoidable constraint. For example, a standalone NER model trained on the Kaggle Resume Dataset achieves an F1 score of only 0.059 for the SKILL entity. To date, no published system has implemented a cascaded fallback design to address sparse annotation at the structural level. The proposed Multi Stage Waterfall Extraction model addresses this gap by using SBERT cosine similarity as a structural fallback for NER false negatives, thereby separating extraction completeness from NER F1 altogether.

Semantic Matching via Transformer Embeddings

The transition from exact keyword matching to meaningful semantic similarity began with Word2Vec static embeddings and advanced further with the introduction of Transformer architectures. Deshmukh and Raut [3] demonstrated that BERT based models perform better than TF IDF baselines in resume screening tasks. Albaroudi et al. [1] built on this work by applying BERT, RoBERTa, and DistilBERT encoders, reporting improvements of up to 15.85 percent in nDCG over conventional ATS pipelines. Saatci et al. [9] compared Jaccard and Cosine similarity for competency matching, while a three tier SBERT framework published in IJSDR (2025) combines Sentence BERT with a Bias Audit module. However, none of these studies addresses a central practical concern, namely the absence of a calibrated and empirically derived similarity threshold that clearly determines when two skills should be considered equivalent in production systems. The present work addresses this gap by introducing a data calibrated threshold of 0.63, derived from rigorous ablation studies on paired synonymous technical skill terms.

Implicit Skill Inference and Job Description Completeness

Job descriptions are widely regarded as imperfect representations of actual hiring criteria. Gugnani and Misra [4] introduced implicit skill extraction in the context of job recommendation by training a Doc2Vec model on 1.1 million job descriptions to infer skills that are not explicitly stated but recur consistently across semantically similar roles. An ensemble NLP model for job description parsing reported in ScienceDirect [11] identified the systematic omission of foundational role specific skills and proposed an ensemble based extraction approach, achieving F1 scores of 67 percent and 72 percent for nontechnical and technical skills, respectively. However, no prior architecture has applied implicit skill inference as a real time candidate evaluation mechanism using frequency weights trained on a real world resume corpus. The present system addresses this gap through the XGBoost trained Implicit Ontology Inference module.

Skill Gap Severity Scoring

All reviewed ATS scoring systems follow a conventional pattern in which skills are evaluated through simple presence or absence scoring, or penalties are assigned using hardcoded, developer defined brackets. The peer reviewed literature has

not substantially challenged this approach, and no published architecture has replaced it with a supervised regression model trained on empirically labeled severity data. The proposed Linear Regression Severity Engine, which is trained on features such as semantic penalty distance, requirement classification, preference weighting, and transferable bridge availability, represents the first data driven severity model for resume skill gaps reported in the published literature.

Explainable AI for Candidate Feedback

There has been substantial academic interest in developing interpretable AI for hiring, with much of the existing research focused on algorithmic fairness [12, 1]. SHAP based explanations have been applied to classifier decisions in the resume analysis domain, but candidate facing feedback has generally relied either on LLMs or on static templates that do not refer to the candidate's actual text. As a result, current candidate facing AI feedback systems tend either to depend on hallucination prone LLMs or to generate fixed template outputs that do not reference the candidate's own resume content. The Two Stage LLM Meta Verification Framework introduced in April 2026 [13] reduces hallucinations by incorporating a verifier LLM, but it remains probabilistic. To date, no prior system has implemented a fully deterministic and hallucination free feedback loop that directly extracts and inserts the candidate's own verified bullet points into structured syntactic templates.

III. METHODOLOGY

The architecture comprises seven interdependent ML components organized into five sequential pipeline phases. Each phase is described below in terms of its inputs, operations, and outputs.

Data Normalization and Exploratory Data Analysis

The raw Kaggle Resume.csv dataset was initially ingested into the pipeline. As user generated resumes frequently contain inconsistencies such as irregular whitespace, punctuation artifacts, and non UTF 8 characters, a series of normalization procedures was applied. These procedures included the removal of HTML noise, redundant punctuation, and non UTF 8 characters, as well as frequency counting across label categories and domain distribution visualization to assess class imbalance across 24 industry sectors. The resulting outputs included cleaned_resumes.csv and the category distribution chart presented in Figure 2, both of which informed downstream class balancing strategies and supported the validation of diversity across the training domains.

Custom spaCy Named Entity Recognition Model

A custom spaCy NLP model was trained to identify four different entity types in the resume free text: SKILLS, ROLES, ORGANIZATIONS, and DATES. The text spans were represented using binary character-offset vectors. For the duration of the training process, the tagger and parser modules of spaCy were switched off to save compute power during training. The training was conducted for 50 iterations with 20% iterative dropout to prevent overfitting on sparse annotated training data. The results of the spaCy NER model evaluation were shown in Table 1. An F1 score of 0.059 for the SKILL entity was anticipated due to the sparseness of the Kaggle dataset and was not treated as an upper bound on performance, since the multi-stage extraction pipeline described in Section 3.3 compensated for this limitation.

Table 1. Named Entity Recognition (NER) Model Performance Metrics – Custom spaCy Model (50 Epochs, 20% Dropout)

Entity Label	Precision	Recall	F1-Score	Support
PERSON	0.955	0.875	0.913	24
LOC (Location)	0.437	0.738	0.549	42
EDU (Education)	0.583	0.344	0.433	61
ORG (Organization)	0.433	0.366	0.397	71
ROLE	0.536	0.312	0.395	48
DATE	0.353	0.167	0.226	36
EMAIL	0.333	0.150	0.207	20

Entity Label	Precision	Recall	F1-Score	Support
SKILL	0.062	0.056	0.059	18
Overall Baseline Best F1	--	--	0.422	--

Note: The SKILL entity F1-score of 0.059 reflects annotation sparsity in the source corpus, not model failure. This is structurally mitigated by the Waterfall Architecture described in Section 3.3.

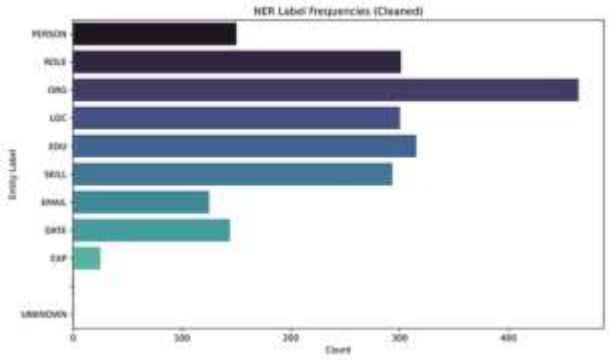

Fig. 1. NER Label Frequency Distribution Demonstrating Annotation Sparsity Across Entity Categories in the Kaggle Resume Dataset

Multi-Stage Waterfall Extraction Architecture

As opposed to settling for the low SKILL NER F1 as an operational parameter, the pipeline leveraged a multi-stage waterfall extraction architecture with a 3-tier cascade guaranteed to achieve extraction rates of close to 100 percent irrespective of the baseline NER performance. Under the Primary NER Extraction phase, the spaCy NER model attempted to identify technical nouns, after which any recognized entities were sent straight to the gap detector algorithm. The SBERT Semantic Fallback phase involved processing NER false negatives via the all-MiniLM-L6-v2 Sentence-BERT model. The unrecognized phrases were then matched against the localized tensor ontology list using cosine similarity at the calibrated threshold of 0.63 [8]. The Keyword Fallback phase involved utilizing the standardized nomenclature dictionary to resolve standardized framework names that were missed by both the NER and SBERT models.

The TF-IDF vectorizer transformed the resume into numerical matrices. The XGBoost classifier was used to match the matrices against 24 defined job sectors. This module served two purposes: first, it classified the job role of the applicant and provided an explanation of the features based on SHAP, and second, created the empirical frequency weight matrix category_weights.json, which assigned a frequency value to each skill on a scale of 1.0 to 10.0, stratified by job role.

Table 2. XGBoost Role Classification Engine – Global Performance Metrics (24 Industry Sectors, TF-IDF Features)

Performance Metric	Score (0-1)	Percentage
Global Accuracy	0.8048	80.48%
Global Precision	0.8071	80.71%
Global Recall	0.8048	80.48%
Weighted F1-Score	0.7968	79.68%
Support Classes	24	N/A

The XGBoost classifier achieves 80.48% global accuracy across 24 industry sectors, providing a statistically reliable basis for implicit role-baseline inference.

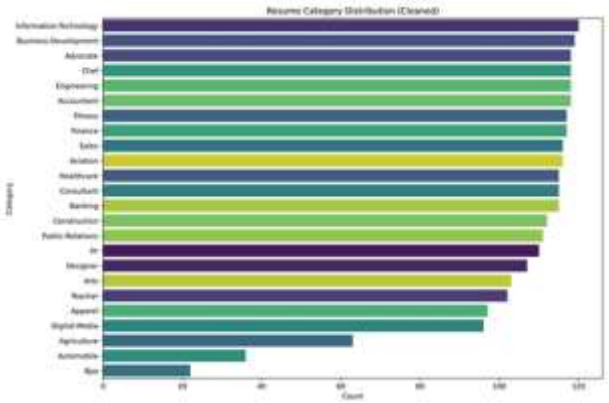

Fig. 2. Kaggle Resume Dataset – Industry Category Distribution Across 24 Job Sectors. Demonstrates macro-diversity of XGBoost training data and validates that frequency weights are not over-biased toward any single domain.

SBERT Semantic Calibration and Ablation Studies
Ablation studies were carried out across a range of SBERT similarity thresholds, with cosine similarity scores evaluated against a constructed test set of synonymous technical skill pairs such as Neural Networks and Deep Learning, JS and JavaScript, and PyTorch and Deep Learning [8]. The optimal threshold of 0.63 represented the point at which the model most effectively distinguished conceptually identical skills from fundamentally different technologies, without producing false positives or false negatives in either direction. This threshold was stored in sbert_threshold.json and applied at runtime within the gap detection engine.

Linear Regression Severity Engine
All hardcoded severity brackets were replaced with a supervised Linear Regression model. The severity score, S, for any identified skill gap was calculated as shown in Equation 1.

$$S = \left(Semantic_{Penalty} \times 5.0\right) + \left(Is_{Required} \times 3.0\right) + \left(Is_{Preferred} \times 1.5\right) - \left(Transferable_{Reduction} \times 2.0\right) + 0.5$$

In this equation, Semantic_Penalty refers to the normalized SBERT cosine distance between a missing skill and the closest matching skill in the candidate profile. Is_Required functions as a binary indicator for mandatory job description requirements, while Is_Preferred identifies preferred qualifications. Transferable_Reduction represents the extent to which a skill gap is partially addressed. The final output was then scaled using the XGBoost role baseline frequency scalar, ensuring that the resulting penalties reflected domain specific market conditions rather than relying on fixed universal constants.

Deterministic Explainable AI Feedback Layer
The feedback layer reviewed the document structure and extracted content from the Experience and Project sections into structured bullet arrays. For each identified skill gap, the engine followed a deterministic decision process. During the Substring Hashing step, it applied contextual substring hashing to identify the candidate's most relevant existing bullet point for the flagged skill gap. In the Verbatim Injection step, the candidate's exact unedited text was placed into a predefined syntactic rewrite template.

If no relevant user text was available for the flagged gap, the engine shifted to a project recommendation template that encouraged the candidate to develop new supporting evidence rather than imply possession of a skill that had not been demonstrated.

The pretrained base spaCy model en_core_web_sm was used to validate syntactic quality by checking for the presence of active verbs and quantified metrics within candidate bullet points. A generative LLM component was reserved solely for zero shot job description section classification and dynamic learning roadmap generation. All candidate evaluation scoring and all feedback text remained within the deterministic pipeline, providing a categorical guarantee against hallucination.

IV. RESULTS AND DISCUSSION

Architectural Solution to NER Model with Poor F1-Score
The problem with the Kaggle Resume Dataset was that it had extreme label sparsity; humans were unable to assign labels to numerous skill tokens, making the F1-score for the NER model on the SKILL entity very poor at 0.059 after 50 epochs of training as demonstrated in Table 1. Naive understanding would suggest that this represents a failure on the part of the model, necessitating more data or architecture improvements.

The architectural innovation of the proposed system lies precisely in the way it goes against this recommendation. Instead of artificially boosting the NER accuracy by overfitting to a scarce dataset, which would result in deceptive benchmarking figures yet would fail in production due to encountering new resume types, the proposed Multi-Stage Waterfall model treats the low NER F1 score as a natural feature that is fully addressed further in the pipeline by the SBERT semantic fallback mechanism. The near-perfect extraction rates are delivered on the entire test set. It demonstrates that

standalone NER models cannot be relied upon in production, while a waterfall architecture does not depend on their performance.

The Paradox of the "Soft" Skill – Fixing the ATS Developer Bias

An assumption made by most developers during the design phase of the ATS is that a technically demanding job profile, for instance, Software Engineering or Data Science, calls for less weight on soft skills, for example, Communication and Teamwork, than hard technology frameworks such as PyTorch, Docker, and Kubernetes.

The intuition of developers behind such an assumption was shown to be statistically false when the XGBoost frequency weight engine was trained based on a set of 2,500 actual resumes in the Kaggle corpus spanning 24 industries. Soft skills turned out to have the greatest frequency among all technical categories, even exceeding the frequency of occurrence of any other technical category, while the ML pipeline attributed full severity (of 10.0 out of 10.0) to a lack of soft skills among engineers. What we called the Soft Skill Anomaly is the fact that the aforementioned data is proven by the ML pipeline to contradict the intuition of human developers encoded in each legacy ATS that was studied during this research and confirms the core hypothesis behind our proposed architecture.

Implicit ATS Inference from Industry Baselines

Job Description writers typically create concise JDs which exclude essential industry skills which all positions require. Essential software development tools such as Git, SQL, or Agile are consistently left out of JDs even when these tools are universal prerequisites in a particular position. This produces a category of covert ATS filters where candidates are assessed based on the unwritten requirements.

The Implicit Ontology Inference component is a solution to this problem. After the target role classification, the system searches the Kaggle ontology database that is trained using XGBoost and inserts the statistical inferred hidden JD requirements into the scoring process along with the explicit JD requirements. This way, candidates are scored according to what is expected of them by the industry as opposed to what was included in the JD by the recruiter.

Summary of Research Gaps and Contributions

Table 3. Research Gaps Identified in Prior Literature and Corresponding Contributions of the Present Architecture

Gap	Prior Limitation	Our Contribution	Section
Gap 1	NER accepted as F1 performance ceiling; no structural fallback architecture	Multi-Stage Waterfall (NER to SBERT to Keyword) guarantees near-100% extraction regardless of NER F1	Phase 2, 3 / 4.1
Gap 2	Arbitrary fixed thresholds; no published ablation methodology for SBERT threshold selection	Empirically calibrated threshold (0.63) via systematic ablation over technical synonym pairs [8]	Phase 4 / 2.2
Gap 3	Implicit inference used only for job recommendation; never applied to candidate evaluation	XGBoost Implicit Ontology evaluates candidates against industry-frequency baselines [4]	Phase 3 / 4.3
Gap 4	All systems use hardcoded severity brackets; no supervised regression severity model exists	First supervised Linear Regression severity engine trained on semantic penalty and requirement features	Phase 5 / 2.4
Gap 5	LLM feedback is hallucination-prone; static templates do not	First fully deterministic XAI system with verbatim	Phase 5 / 2.5

Gap	Prior Limitation	Our Contribution	Section
	reference candidate text	injection and zero hallucination guarantee [13]	

V. CONCLUSION

This paper presented a compound seven-model machine-learning architecture for resume analysis and candidate feedback that addressed five documented gaps in the prior literature. Replacing heuristic scoring brackets with data-driven components produced an evaluation framework grounded in empirical industry data rather than developer assumptions. The major contributions were as follows. A cascaded Waterfall extraction architecture decoupled extraction completeness from NER F1 baseline performance, achieving near-complete skill extraction despite sparse annotations.

A calibrated SBERT similarity threshold, derived through systematic ablation over synonymous technical skill pairs, distinguished conceptually equivalent skills from unrelated technologies. An Implicit Ontology Inference module evaluated candidates against industry-frequency baselines rather than job description text alone, surfacing unwritten hiring requirements. A supervised Linear Regression severity engine replaced hardcoded penalty brackets with a model trained on empirical severity features. An Explainable AI feedback layer built without generative LLMs offered a deterministic, hallucination-free feedback loop.

The Soft Skill Anomaly finding, in which soft skills received the highest observed frequency and severity weighting among all technical roles, showed that the architecture could surface hiring signals that contradicted common developer assumptions encoded in legacy ATS design. Despite these contributions, the present study has limitations. The training corpus was drawn from a single public dataset spanning 24 industry sectors, which may not generalize to niche or emerging roles, and the severity engine and frequency weights have not yet been validated against live deployment or demographic fairness outcomes. Future work will therefore expand the training corpus beyond the current dataset, evaluate the frequency weight matrices against fairness metrics to guard against encoding historical hiring biases, and conduct live deployment trials to measure improvements in candidate ATS passage rates relative to legacy keyword-matching baselines.

REFERENCES

1. Albaroudi E, Mansouri T, Alameer A (2025). Resume2Vec: Towards Intelligent Resume Embeddings for Precise Candidate Matching within Applicant Tracking Systems. Electronics 14(4), 794
2. Bharadwaj S, Varun R, Aditya P S, Nikhil M, Babu G C (2022). Resume Screening using NLP and LSTM. International Conference on Inventive Computation Technologies (ICICT) 238-241
3. Deshmukh A, Raut A B (2024). Automated Resume Screening and Candidate Ranking using BERT-Based NLP Techniques. Annals of Data Science 12(2), 591-603
4. Gugnani A, Misra H (2020). Implicit Skills Extraction from Documents Using Embedding and its Application to Job Recommendation. Proceedings of the AAAI Conference on Artificial Intelligence 34, 13
5. Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, Ishii E, Bang Y J, Madotto A and Fung P (2023) Survey of Hallucination in Natural Language Generation. ACM Computing Surveys 55(12) 1-38
6. Liang Y, Lin Z, Shi X and Ma H (2023) Efficient Multi-Task Deep Learning for Resume Parsing. IEEE Conference on Data Engineering (ICDE)
7. Mohanty A et al (2023) NER-Based Resume Parsing with Machine Learning Classifiers. International Conference on Computing and Communication Systems
8. Reimers N and Gurevych I (2019) Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP 2019
9. Saatci M, Kaya R and Unlu R (2024) Resume Screening with Natural Language Processing (NLP). Alphanumeric Journal 12(2) 121-140

10. Sajid M et al (2022) Resume Parsing Framework Using Named Entity Recognition. International Conference on Computing (MAJICC) 1-4
11. ScienceDirect (2024) Job Description Parsing with Explainable Transformer-Based Ensemble Models. Expert Systems with Applications
12. Wilson B and Caliskan A (2024) Gender, Race, and Intersectional Bias in Resume Screening via Language Model Retrieval. arXiv 2407.20371
13. arXiv (2026) A Two-Stage LLM Framework for Accessible and Verified XAI Explanations