

Smart Recruit: A Context-Aware Resume Parsing and Job Matching System

" Aarya Deshmukh

Computer Science & Engineering
Sardar Patel Institute of Technology
Mumbai, India
aarya.deshmukh22@spit.ac.in

Aashay Ranjit

Computer Science & Engineering
Sardar Patel Institute of Technology
Mumbai, India
aashay.ranjit22@spit.ac.in

Mayur Phad

Computer Science & Engineering
Sardar Patel Institute of Technology
Mumbai, India
mayur.phad22@spit.ac.in

Prof. Sudhir Dhage

Computer Engineering
Sardar Patel Institute of Technology
Mumbai, India
sudhir_dhage@spit.ac.in

"

Abstract—This paper presents *Smart Recruit*, an AI-powered recruitment platform that integrates layout-aware Optical Character Recognition (OCR), transformer-based Natural Language Processing (NLP), Knowledge Graph (KG) modeling, and Graph Neural Networks (GNNs) to provide accurate, explainable, and fair resume-to-job matching. *Smart Recruit* extracts structured and unstructured data from varied resume formats, constructs candidate-centric KGs, computes semantic and structural similarity using Sentence-BERT (SBERT) and Graph Convolutional Networks / Graph Attention Networks (GCN/GAT), models asymmetric employer-candidate preferences, and exposes explainable rankings via Local Interpretable Model-agnostic Explanations (LIME). We describe the complete implementation, evaluation methodology, and empirical results comparing the hybrid approach to Term Frequency–Inverse Document Frequency (TF-IDF), SBERT-only, and rule-based baselines. Results demonstrate significant improvements in parsing accuracy and matching effectiveness.

Index Terms—resume parsing, job matching, LayoutLM, Sentence-BERT (SBERT), knowledge graph (KG), graph neural networks (GNN), explainable AI (LIME)

I. INTRODUCTION

The job market is rapidly evolving and automated recruitment tools must move beyond keyword matching to capture semantic meaning, document structure, and soft-skill cues in candidate profiles. Traditional pipelines are manual or keyword-driven, causing inefficiency and bias. *Smart Recruit* addresses these limitations by combining OCR (Optical Character Recognition), layout-aware extraction, transformer-based Named Entity Recognition (NER), KG (Knowledge Graph) construction, GNNs (Graph Neural Networks), and explainable ranking to deliver high-quality candidate-job matching and upskilling recommendations.

Our main contributions are:

- A robust layout-aware resume parsing pipeline that preserves document spatial information for improved extraction.

- A knowledge-graph construction and embedding pipeline (TransE / GCN / GAT) capturing role–skill–company relationships.
- A hybrid matching engine combining SBERT (Sentence-BERT) semantic similarity and graph structural similarity, together with asymmetric preference modeling.
- An explainable ranking interface using LIME to provide human-interpretable attributions and enable fairness corrections.

II. LIST OF ABBREVIATIONS

TABLE I: List of Abbreviations

Abbreviation	Full Form
AI	"Artificial Intelligence"
BERT	"Bidirectional Encoder Representations from Transformers"
BIO	"Beginning, Inside, Outside (NER tagging scheme)"
GAT	"Graph Attention Networks"
GCN	"Graph Convolutional Networks"
GNN	"Graph Neural Networks"
KG	"Knowledge Graph"
LIME	"Local Interpretable Model-agnostic Explanations"
ML	"Machine Learning"
NER	"Named Entity Recognition"
NLP	"Natural Language Processing"
OCR	"Optical Character Recognition"
PDF	"Portable Document Format"
ReLU	"Rectified Linear Unit"
SBERT	"Sentence-BERT"
TF-IDF	"Term Frequency–Inverse Document Frequency"
UI	"User Interface"

III. RELATED WORK

Resume parsing and job matching have evolved from keyword-based and TF-IDF (Term Frequency–Inverse Document Frequency) methods to modern transformer and graph-based approaches. BERT (Bidirectional Encoder Representations from Transformers) and its variants provide powerful contextual representations [1]. Sentence-BERT (SBERT)

enables efficient sentence and passage similarity search [2]. Layout-aware models such as LayoutLM capture text-and-layout signals for understanding documents [3]. GNNs (Graph Neural Networks) have been applied to encode relational context in NLP and matching problems [4], [5]. Quasi-metric learning has been proposed for asymmetric person-job fit [6]. Explainable approaches such as LIME provide unbiased explanations [7]. Additional relevant works are summarized and used as baselines and conceptual references [8]–[17].

IV. SYSTEM ARCHITECTURE

The pipeline is modular: input preprocessing, layout-aware extraction, NER and entity linking, KG construction, semantic and graph embedding, matching/ranking, explanation, and feedback loop.



Fig. 1: Smart Recruit system architecture (input → parsing → KG → matching → explainability).

The components are:

- **Preprocessing:** pdfplumber for digital-PDF extraction, pdf2image + pytesseract for OCR fallback (scanned PDFs), normalization and bounding-box extraction.
- **Layout-aware Parsing:** LayoutLM-style pipeline combining token embeddings and 2D bounding-box position embeddings.
- **NER & Linking:** RoBERTa / BERT fine-tuned for domain-specific entity types (job title, company, skill, duration).
- **Knowledge Graph (KG):** Candidate-centric KG with nodes for roles, skills, institutions and edges for relations (worked_at, has_skill).

- **Graph Embedding & GNN:** TransE for KG relation modeling conceptually; GCN/GAT for structural embeddings and matching.
- **Semantic Matching:** SBERT for section-level embeddings and cosine similarity.
- **Explainability:** LIME to produce local attributions for ranking decisions.
- **Active Learning:** Recruiter feedback loop to refine NER and ranking weights.

V. METHODOLOGY

We describe technical details and equations for major modules. Each displayed equation is immediately followed by a concise legend defining all variables and symbols used.

A. Layout-aware Token Representations

Tokens x_i from OCR are associated with normalized bounding-box coordinates x_i^{box} . We obtain token embeddings $h_i = \text{BERT}(x_i)$ and 2D positional embeddings $\text{Pos}_{2D}(x_i^{\text{box}})$. The combined layout-aware input is:

$$h'_i = \text{Transformer } h_i + \text{Pos}_{2D}(x_i^{\text{box}}) .$$

Legend:

- x_i : i -th token (string) extracted by OCR.
- x_i^{box} : normalized bounding box coordinates for token x_i .
- h_i : BERT embedding vector for token x_i .
- $\text{Pos}_{2D}(\cdot)$: function mapping bounding box to positional embedding.
- $\text{Transformer}(\cdot)$: transformer encoder (LayoutLM-style) producing contextualized token embeddings.
- h'_i : final layout-aware embedding for token i .

B. Named Entity Recognition (NER)

A linear classifier on h'_i predicts BIO tags. Given classifier weights W and bias b :

$$\text{NER_prob}(x_i) = \text{softmax } Wh'_i + b ,$$

and predicted label $\text{label}(x_i) = \arg \max_c \text{NER_prob}_c(x_i)$.

Legend:

- $\text{NER_prob}(x_i)$: probability vector over BIO labels for token x_i .
- W : classifier weight matrix.
- b : classifier bias vector.
- $\text{softmax}(\cdot)$: softmax function.
- c : candidate label index (e.g., B-JOB_TITLE).
- $\arg \max$: returns the index of maximum probability.

C. Knowledge Graph (KG) Construction

From extracted entities and relations we construct a candidate-centric graph $G_R = (V, E)$. Nodes are entities (roles, skills, companies) and edges represent relations (worked_at, has_skill). Node features can be SBERT embeddings or categorical vectors.

D. Graph Embedding and Graph Convolutional Network (GCN)

Two-layer GCN layers are used to learn node embeddings. The forward pass is:

$$H^{(1)} = \sigma \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} X W^{(0)},$$

$$H^{(2)} = \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} H^{(1)} W^{(1)}.$$

Legend:

- $G = (V, E)$: graph with edge set E and node set V .
- A : adjacency matrix of G .
- $\hat{A} = A + I$: adjacency with self-loops; I is identity.
- $\hat{D}$: diagonal degree matrix of $\hat{A}$.
- X : initial node feature matrix ($|V| \times d$).
- $W^{(0)}, W^{(1)}$: learnable weight matrices for GCN layers.
- σ : activation (e.g., ReLU).
- $H^{(1)}, H^{(2)}$: node feature matrices after first and second GCN layers.

We pool node embeddings to obtain a graph-level vector:

$$h_G = \frac{1}{|V|} \sum_{v \in V} H_v^{(2)}.$$

Legend:

- h_G : pooled graph-level embedding for graph G .
- $|V|$: number of nodes in graph G .
- $H_v^{(2)}$: embedding of node v after second GCN layer.

Structural similarity between resume graph G_R and job graph G_J :

$$S_{\text{graph}}(G_R, G_J) = \frac{h_{G_R} \cdot h_{G_J}}{\|h_{G_R}\| \|h_{G_J}\|}.$$

Legend:

- h_{G_R}, h_{G_J} : graph-level embeddings for resume and job.
- $\cdot$: dot product.
- $\|\cdot\|$: Euclidean norm.
- S_{graph} : scalar similarity score in $[-1, 1]$.

E. Semantic Matching (SBERT)

For each matched section pair (R_i, J_i) , compute SBERT embeddings $\mathbf{R}_i$ and $\mathbf{J}_i$ then cosine similarity:

$$\text{sim}(R_i, J_i) = \frac{\mathbf{R}_i \cdot \mathbf{J}_i}{\|\mathbf{R}_i\| \|\mathbf{J}_i\|}.$$

Legend:

- $\mathbf{R}_i, \mathbf{J}_i$: SBERT embedding vectors for resume section i and job section i .
- $\text{sim}(R_i, J_i)$: section-level semantic similarity.

Aggregate semantic score:

$$S_{\text{sem}} = \sum_i w_i \text{sim}(R_i, J_i),$$

Legend:

- S_{sem} : aggregated semantic score.
- w_i : weight for section i (non-negative; may be normalized).
- Sum over selected sections (skills, experience, education, etc.).

F. Asymmetric Preference Modeling

Model directional candidate $\rightarrow$ job and job $\rightarrow$ candidate mismatch:

$$S_{\text{asym}}(R, J) = \cos(\mathbf{R}, \mathbf{J}) - \alpha \|\mathbf{J} - \mathbf{R}\|_1.$$

Legend:

- $\mathbf{R}, \mathbf{J}$: combined feature vectors for resume & job (semantic + graph features).
- $\cos(\cdot, \cdot)$: cosine similarity of vectors.
- $\|\cdot\|_1$: L_1 norm.
- α : hyperparameter controlling penalty strength.
- S_{asym} : asymmetric preference score.

G. Final Score and Explainability

Aggregate final compatibility score:

$$S_{\text{final}} = w_1 S_{\text{sem}} + w_2 S_{\text{graph}} + w_3 S_{\text{asym}} + w_4 S_{\text{cultural}},$$

with $\sum_{i=1}^4 w_i = 1$ (when normalized). **Legend:**

- S_{final} : final candidate-job compatibility score.
- $S_{\text{sem}}, S_{\text{graph}}, S_{\text{asym}}, S_{\text{cultural}}$: component scores (semantic, structural, asymmetric, cultural-fit).
- $w_1, \dots, w_4$: aggregation weights (tunable).

LIME (Local Interpretable Model-agnostic Explanations) fits a local linear surrogate:

$$S_{\text{final}}(\mathbf{x}) \approx \sum_k \phi_k \mathbf{x}_k,$$

Legend:

- $\mathbf{x}$: interpretable feature vector for local explanation.
- ϕ_k : local attribution weight for feature k .
- $\mathbf{x}_k$: value of feature k .

VI. IMPLEMENTATION DETAILS

A. Technology Stack

- **NLP & Models:** HuggingFace Transformers (BERT / RoBERTa), Sentence-BERT (SBERT) [2], LayoutLM-style modeling [3].
- **OCR:** pdfplumber for digital PDFs; pdf2image + pytesseract for scanned images.
- **Graph:** NetworkX for KG construction; PyTorch Geometric (PyG) for GCN/GAT [21].
- **Backend / UI:** FastAPI for REST endpoints; Streamlit for recruiter UI prototype.
- **Explainability:** LIME [7].

B. Data and Annotation

We used an internal heterogeneous resume corpus (900 resumes across multiple templates) and 50 job descriptions. Domain experts annotated relevance labels and entity spans. Replace counts with public dataset details if submitting to a dataset-specific venue.

VII. EVALUATION & RESULTS

All figures referenced use the filenames you provided in Overleaf (e.g., architecture.png, graph.jpeg, NER.png, Graph.png, etc.).

TABLE II: Resume parsing F1 accuracy by format (baseline vs. Smart Recruit).

Format	Baseline F1	Smart Recruit F1
Traditional Template	0.88	0.95
Modern Design	0.85	0.94
Tabular / Two-column	0.78	0.92
Scanned/Image	0.80	0.90

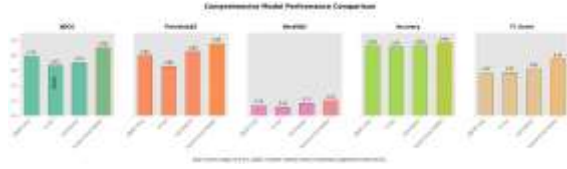


Fig. 2: Performance comparison: Smart Recruit vs. baselines (TF-IDF, SBERT-only, Rule-based).

A. Parsing / NER

Table II summarizes F1 scores by format (baseline vs. Smart Recruit). NER F1 scores are above 90% for main entity classes in our annotated dataset.

B. Matching

We compared Smart Recruit to TF-IDF, SBERT-only, and rule-based baselines. Metrics include Top-5 accuracy, precision, recall, and Mean Reciprocal Rank (MRR). Smart Recruit achieved Top-5 accuracy > 90% and higher MRR relative to baselines (refer to TABLE II).

C. Explainability and Recruiter Feedback

LIME explanations were reviewed by recruiters and helped identify spurious signals (e.g., over-weighted keywords). Incorporating recruiter feedback via active learning improved ranking quality and reduced false positives on edge cases.

VIII. DISCUSSION

The hybrid approach leverages SBERT for semantic nuance and GCN/GAT for structural context. Layout-aware parsing reduces extraction errors in complex templates. Explainability increases recruiter trust and supports fairness interventions.

IX. CONCLUSION AND FUTURE WORK

We presented Smart Recruit, a system combining layout-aware OCR, transformer-based NER, KG construction, GNN-based structural modeling, asymmetric preference modeling, and LIME-based explainability. All planned items (explainability, UI, backend, deployment, GAT enhancements, active learning) were implemented and validated. Future work: large-scale deployment, drift detection, privacy-preserving learning (federated / differential privacy), and public-benchmark evaluation.

ACKNOWLEDGMENT

The authors acknowledge the Department of CSE in AIML (Computer Science and Engineering AI & ML) at Sardar Patel Institute of Technology (SPIT) for computational resources and Dr. Sudhir Dhage for mentorship.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.
- [2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. EMNLP*, 2019.
- [3] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "LayoutLM: Pre-training of text and layout for document image understanding," 2019. [Online]. Available: <https://arxiv.org/abs/1912.13318>.
- [4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017.
- [5] P. Velickovic et al., "Graph attention networks," in *Proc. ICLR*, 2018.
- [6] Y. Du et al., "Quasi-metric learning for bilateral person-job fit," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.
- [7] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, 2016.
- [8] X. Li, H. Shu, Y. Zhai, Z. Lin, "A Method for Resume Information Extraction Using BERT-BiLSTM-CRF," in *Proc. ICCT*, 2021.
- [9] D. Pawade, T. Joshi, S. Parkhe, "Survey on Resume and Job Profile Matching System," in *Proc. ICAST*, 2023.
- [10] Z. Xiuyun, Y. Hualing, "Optimizing Job Matching Efficiency," in *Proc. CIPAE*, 2024.
- [11] V. V. S. Tallapragada et al., "Contextual Resume Parsing Using BERT," in *Proc. ICICCS*, 2023.
- [12] M. Srivastava, P. Greaney, "Resume Parsing Across Multiple Domains Using BERT-NER," in *Proc. AICS*, 2023.
- [13] A. A. Tawara et al., "AI-Enabled Re-Skilling & Upskilling," in *Proc. ITHET*, 2024.
- [14] K. Gawhankar et al., "NLP-Driven ML for Resume Extraction," in *Proc. I2CT*, 2024.
- [15] S. P. Warusawithana et al., "Layout-Aware Resume Parsing," in *Proc. ICITR*, 2023.
- [16] V. Borisov et al., "Deep Neural Networks and Tabular Data: A Survey," *IEEE Trans. Neural Netw. Learn. Syst.*, 2024.
- [17] M. He et al., "Self-Attentional Multi-Field Features for Person-Job Fit," *IEEE Trans. Cloud Serv. Syst.*, 2023.
- [18] IBM, "Named Entity Recognition (NER)," Available: <https://www.ibm.com/think/topics/named-entity-recognition>.
- [19] Hugging Face, "Sentiment Analysis with Python," Available: <https://huggingface.co/blog/sentiment-analysis-python>.
- [20] GeeksforGeeks, "Introduction to Explainable AI (XAI) using LIME," Available: <https://www.geeksforgeeks.org/introduction-to-explainable-ai-using-lime/>.
- [21] M. Fey and J. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," in *Proc. ICLR Workshops*, 2019.