*Article*

# Transformer-Based Semantic Embedding Model for Resume-Job Matching in Intelligent Talent Screening

**Yuerong Yan** [1,*]

[1] Shanghai Zizen Consulting Co., Ltd., Shanghai, China
* Correspondence: Yuerong Yan, Shanghai Zizen Consulting Co., Ltd., Shanghai, China

**Abstract:** Based on the semantic matching requirements between resume text and job descriptions, this study investigates the application of Transformer semantic embedding models in intelligent talent screening. By constructing dual-sided semantic encoding networks for resumes and job postings, we design a dual-tower embedding matching structure and semantic scoring mechanism to achieve unified semantic representation and matching ranking between candidates and job requirements. Experiments validated model performance using real recruitment datasets, with ablation studies analyzing contributions from different semantic features. Results show the model achieves 89.47% accuracy, 88.63% recall, and an F1 score of 89.04%. Removing job constraint semantics reduces accuracy to 84.58%, demonstrating that integrating semantic embedding with constraint fusion significantly enhances recruitment matching effectiveness.

**Keywords:** Transformer; Resume-Job Matching; Semantic Embedding; Dual-Tower Model; Talent Screening

Received: 03 February 2026
Revised: 22 March 2026
Accepted: 06 April 2026
Published: 10 April 2026

## 1. Introduction

Internet recruitment platforms continuously accumulate massive volumes of resume and job description texts. The screening efficiency and matching precision of candidates have gradually become critical factors influencing the quality of recruitment decisions. Traditional keyword-based or rule-based matching methods rely on shallow textual features, struggling to capture deep semantic connections between candidates' experiences, skill structures, and job requirements. This results in significant limitations in matching accuracy and recommendation reliability. Addressing this challenge, researchers have progressively integrated natural language processing and deep learning technologies into recruitment matching tasks to enhance semantic understanding and talent screening efficiency.

Existing research explores intelligent recruitment matching methods through diverse technical approaches. A resume-to-job matching system based on BERT semantic understanding has been proposed, leveraging deep semantic representations to improve recruitment recommendation performance [1]. Natural language processing and machine learning techniques have also been employed to refine job recommendation models, emphasizing the critical role of textual semantic features in recruitment systems [2]. The BERT model has further been applied to automated resume screening and candidate ranking, demonstrating that deep semantic modeling can effectively improve candidate screening efficiency [3]. An ontology-based recruitment matching approach has also been proposed from the perspective of skill semantic structure, achieving structured associations between job requirements and skill elements [4]. In addition, a systematic review of BERT's semantic representation capabilities in natural language processing has highlighted the significant advantages of the Transformer architecture in contextual semantic modeling [5].

Despite progress in related research, existing methods exhibit several limitations. Some studies rely on single semantic representation models, struggling to capture multidimensional semantic relationships between resume text and job descriptions. Furthermore, the matching process between candidates and positions lacks a unified semantic embedding space, leaving room for improvement in matching efficiency and ranking accuracy. To address these issues, this paper constructs a resume-to-job matching model based on Transformer semantic embeddings. It employs a dual-tower semantic encoding architecture to achieve unified semantic representations of resume text and job descriptions. Additionally, it designs a semantic similarity scoring and candidate ranking strategy to enhance the matching accuracy and recommendation efficiency of intelligent talent screening.

## 2. Resume-Job Semantic Feature Modeling and Data Processing

Before undergoing semantic encoding, resume text and job description text must undergo unified feature normalization. Otherwise, semantic misalignment may occur between job competency items, experience fragments, and role constraints due to differences in expression granularity. The data processing module first performs field segmentation and standardized mapping for educational background, skill terms, project experience, job responsibilities, and qualification requirements. It then constructs sample quality constraints based on terminology consistency, entity completeness, and temporal continuity, as shown in Table 1. Different fields retain structural information such as academic level hierarchy, skill phrase density, and responsibility verb intensity [6]. The initial representation of the semantic feature vector can be written as:

**Table 1.** Structured Text Features for Resume-Position Pairing.

| Feature Category | Core Resume Fields | Core Job Field | Processing Objective |
|---|---|---|---|
| Basic Attributes | Education Level, Major, Years of Experience | Education Requirements, Major Restrictions, Experience Requirements | Unified Hierarchy Standards |
| Skill Semantics | Tools, Languages, Certifications, Professional Skills | Essential Skills, Preferred Skills | Extracted skill phrases |
| Experience Semantics | Project names, responsibilities, outcomes | Job responsibilities, scope of tasks | Align job responsibilities semantics |
| Chronological Information | Tenure Start/End Dates, Project Duration | Onboarding Timeline, Years of Experience | Establishing Time Constraints |

$$z_i=[\alpha_i e_i^s; \beta_i e_i^x; \gamma_i e_i^t] \qquad (1)$$

where $\mathbf{z}_i$ denotes the fused representation of the $i_{th}$ sample fragment, $\mathbf{e}_i^s$ represents the skill semantic embedding, $\mathbf{e}_i^x$ denotes the experience semantic embedding, $\mathbf{e}_i^t$ indicates the temporal constraint embedding, and $\alpha_i$ , $\beta_i$ , and $\gamma_i$ denote the corresponding weights. The data filtering stage further employs a consistency scoring function:

$$Q_i=\mu_1 r_i^c + \mu_2 r_i^d + \mu_3 r_i^u \qquad (2)$$

Where $Q_i$ denotes sample quality score, $r_i^c$ denotes field completeness rate, $r_i^d$ denotes retention density after semantic denoising, $r_i^u$ denotes field unit standardization rate, and $\mu_1$ , $\mu_2$ , $\mu_3$ denote weighting coefficients. This processing outcome directly determines representation stability and matching score reliability in the subsequent dual-tower semantic space.

## 3. Transformer-Based Resume-Position Semantic Embedding Model

### 3.1. Transformer Semantic Encoding Mechanism

The Transformer semantic encoding mechanism maps standardized resume fragments and job position fragments into context-relevant representations. Therefore, the encoding process must simultaneously preserve cross-word dependencies among skill entities, experience logic, and job constraints. As shown in Figure 1, the input sequence first forms a composite representation by combining word vectors, position vectors, and field type vectors. It then enters a multi-head self-attention layer to capture long-range semantic associations.The unidirectional attention output can be expressed as:

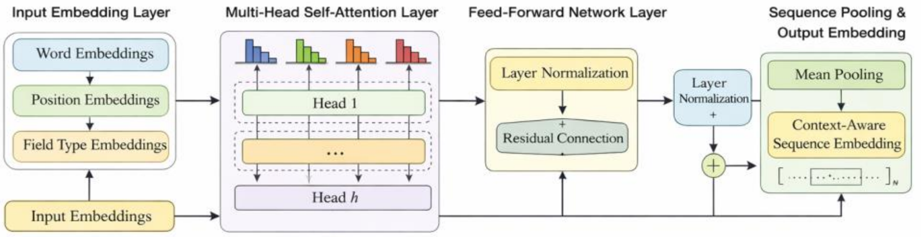

Figure 1 Schematic of Transformer Semantic Encoding Mechanism

$$\mathrm{A}^{(\mathrm{h})} = \mathrm{softmax}\left(\frac{\mathrm{Q}^{(\mathrm{h})}{\mathrm{K}^{(\mathrm{h})}}^{\mathrm{T}}}{\sqrt{\mathrm{d}_{\mathrm{k}}}}\right)\mathrm{V}^{(\mathrm{h})} \quad (3)$$

where $\mathrm{A}^{(\mathrm{h})}$ denotes the output of the h th attention head, $\mathrm{Q}^{(\mathrm{h})}$ , $\mathrm{K}^{(\mathrm{h})}$ , and $\mathrm{V}^{(\mathrm{h})}$ represent the query matrix, key matrix, and value matrix respectively, and $d_k$ denotes the key vector dimension. The multi-head results undergo concatenation and a feedforward transformation to yield the context encoding H , whose aggregated representation is expressed as:

$$\mathrm{u}=\mathrm{LayerNorm}(\mathrm{MeanPool}(\mathrm{H}) + \mathrm{b}_{\mathrm{u}}) \; (4)$$

where u denotes the sequence-level semantic embedding, $\mathrm{MeanPool}(\cdot)$ represents the average pooling operator, and $\mathrm{b}_u$ denotes the bias term. This embedding serves as the direct input for subsequent bilateral semantic alignment and similarity calculation.

### 3.2. Design of the Resume Text Semantic Encoding Network

The resume text semantic encoding network requires a hierarchical representation mechanism tailored to the heterogeneous structure of candidate experiences, as educational background, project experience, skill certifications, and employment trajectories exhibit significant differences in semantic density and discriminative contribution.The encoding network first segments the input sequence into educational units, skill units, and experience units based on field boundaries. It then constructs candidate-side initial representations through shared word embeddings and field gating vectors, preserving the structural origins of different resume fragments. The field-level semantic aggregation result can be expressed as:

$$\mathrm{r}_{\mathrm{m}}=\sum_{\mathrm{n}=1}^{\mathrm{L}_{\mathrm{m}}} \omega_{\mathrm{mn}}\, \mathrm{h}_{\mathrm{mn}},\omega_{\mathrm{mn}}=\frac{\exp(\mathrm{c}_{\mathrm{m}}^{\mathrm{T}}\mathrm{h}_{\mathrm{mn}})}{\sum_{\mathrm{t}=1}^{\mathrm{L}_{\mathrm{m}}} \exp(\mathrm{c}_{\mathrm{m}}^{\mathrm{T}}\mathrm{h}_{\mathrm{mt}})}(5)$$

Among these, $\mathrm{r}_m$ represents the aggregated vector of the $m$ th resume field, $L_m$ denotes the number of tokens corresponding to this field, $\mathrm{h}_{mn}$ indicates the hidden state of the $n$ th token after contextual encoding, $\omega_{mn}$ signifies the attention weights, and $\mathrm{c}_m$ denotes the semantic query vector for the field. The overall candidate representation is further obtained by concatenating the aggregated vectors of each field and applying a linear projection, as shown in Table 2. Project responsibilities and skill phrases typically exhibit higher discriminative power in ranking tasks, leading the network to form more concentrated weight distributions for these fields during training. In contrast, work experience and education level primarily serve as boundary constraints. This division directly influences the subsequent semantic alignment trajectory between candidates and job requirements [7].

**Table 2.** Semantic Features of Resume Fields and Their Roles in Candidate Representation.

| Field Category | Primary Information Source | Semantic Representation Format | Function in Candidate Semantic Representation | Impact on Matching Score |
|---|---|---|---|---|
| Educational Background | Academic Level, Major Field, Graduating Institution | Educational Level Embedding + Major Semantic Vector | Provides foundational capability boundaries for candidates | Determines the degree to which the position's educational and major requirements align with the candidate's qualifications |
| Skill Information | Professional skills, software tools, technical certifications | Semantic Embedding of Skill Phrases | Describe the structure of a candidate's professional capabilities | Enhance the accuracy of job skill requirement matching |
| Project Experience | Project Name, Role Assumed, Technical Solution | Project semantic vector + role weight vector | Reflecting practical capabilities and depth of technical application | Enhance semantic relevance of job responsibilities |
| Work Experience | Position Held, Years of Experience, Responsibilities Description | Experience semantic vector + time series features | Describe career trajectory and experience accumulation | Influencing Experience Requirement Matching Scores |
| Responsibilities & Outcomes | Project Outcomes, Performance Metrics, Technical Contributions | Semantic representation of outcomes + intensity weighting | Demonstrates candidate's actual output capability | Enhance identification of high-value candidates |

### *3.3. Design of the Job Description Semantic Encoding Network*

The semantic encoding network for job descriptions requires precise semantic modeling of role requirements and competency constraints within job texts. This is because job descriptions typically incorporate multiple information structures-core tasks, competency thresholds, and supplementary conditions-each carrying distinct weighting during candidate screening. Job texts are first segmented into semantic chunks based on responsibility statements, skill phrases, and experience requirements. Subsequently, a Transformer encoding layer extracts context-aware latent state sets for each role: $\mathbf{G}$={$\mathbf{g}_1$,$\mathbf{g}_2$,…,$\mathbf{g}_p$} [8]. The overall semantic representation of the job is formed through attention-weighted aggregation:

$$\mathrm{v_p}=\sum_{\mathrm{q=1}}^{\mathrm{P}}\eta_{\mathrm{q}}\,\mathrm{g_q},\ \eta_{\mathrm{q}}=\frac{\exp\left(\mathrm{w_p^T g_q}\right)}{\sum_{\mathrm{s=1}}^{\mathrm{P}}\exp\left(\mathrm{w_p^T g_s}\right)}\quad(6)$$

where $\mathbf{v}_p$ denotes the semantic embedding vector of the job description, $P$ represents the total number of tokens in the job text, $\eta_q$ indicates the semantic weight of the $q_{th}$ token, and $\mathrm{w}_p$ signifies the job semantic discrimination parameter. The strength of job constraints is further modeled across three dimensions: responsibility density, experience threshold, and skill rigidity:

$$\delta_{\mathrm{p}} = \varrho_1\mathrm{m_p^r} + \varrho_2\mathrm{m_p^e} + \varrho_3\mathrm{m_p^k} \qquad (7)$$

where $\delta_p$ denotes the position constraint score, $m_p^r$ represents the responsibility density metric, $m_p^e$ indicates the experience threshold metric, $m_p^k$ signifies the skill rigidity metric, and $\rho_1$,$\rho_2$,$\rho_3$ is the weight coefficient. During training, both position semantic representation and constraint scoring must participate in gradient updates. Therefore, the encoding phase typically requires implementing attention weight calculation and constraint factor fusion, as illustrated below:

### 3.3.1. Position semantic encoding

```
hidden_states = transformer_encoder(position_tokens, attention_mask = mask)
attention_weights = torch.softmax(torch.matmul(hidden_states, wp), dim = 1)
position_vector = torch.sum(attention_weights * hidden_states, dim = 1)
constraint_score = r1 * duty_density + r2 * exp_requirement + r3 * skill_rigidity
```

## 4. Dual-Tower Semantic Embedding Matching Structure Construction

The dual-tower semantic embedding matching structure requires balancing parameter independence with semantic comparability. Candidate resumes emphasize experience expansion and skill accumulation, while job postings highlight duty boundaries and constraint intensity. These text types exhibit asymmetry in syntactic density and discriminative focus. As shown in Figure 2, the matching structure feeds candidate-side encoded vectors and job-side encoded vectors into independent projection layers, then performs distance constraints and similarity calculations through a shared semantic space. This design mitigates semantic interference from direct concatenation of heterogeneous texts while preserving independent discriminative capabilities for both sides [9]. The matching score in the unified embedding space is defined as:

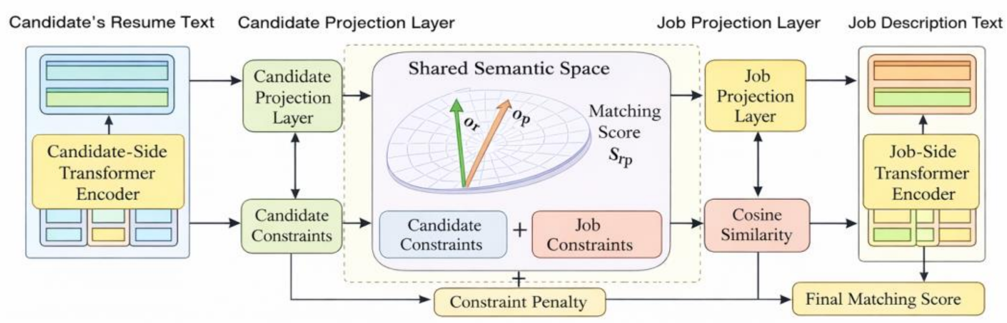

Figure 2 Schematic Diagram of the Twin-Tower Semantic Embedding Matching Structure

$$\mathrm{S_{rp}}=\frac{\mathrm{o_r^T\,o_p}}{\|\mathrm{o_r}\|_2\|\mathrm{o_p}\|_2}-\lambda_{\mathrm{c}}\left|\delta_{\mathrm{p}}-\kappa_{\mathrm{r}}\right| \quad (8)$$

where $S_{rp}$ denotes the matching score between candidate $r$ and position $p$ , $\mathrm{o}_r$ represents the candidate projection vector, $\mathrm{o}_p$ denotes the position projection vector, $\|\cdot\|_2$ indicates the L2 norm, $\delta_p$ signifies the position constraint strength, $\kappa_r$ reflects the candidate competency fulfillment, and $\lambda_c$ represents the constraint deviation penalty coefficient. During training, highly correlated sample pairs are pulled closer while weakly correlated pairs are pushed farther apart. Concurrently, constraint penalties suppress mismatches where "semantically close but ineligible" candidates are matched. This process generates matching trajectories directly supporting candidate ranking and recall evaluation.

## 5. Resume-Position Semantic Matching Scoring Mechanism

The semantic matching scoring mechanism must convert similarity relationships in the embedding space into quantifiable, sortable results. This is because the candidate screening process considers not only textual semantic proximity but also the overall alignment between competency fit, experience relevance, and role boundaries. Upon receiving candidate vectors and job vectors, the scoring mechanism does not directly employ a single similarity metric as the final output. Instead, it incorporates semantic proximity, constraint fulfillment, and field alignment strength into a discriminative function. This approach reduces the probability of samples with "keyword similarity but mismatched responsibilities" ranking highly. The composite matching score is defined as:

$$\mathrm{M}_{\mathrm{rp}} = \sigma\left(\theta_1 \cos\left(\mathrm{o}_{\mathrm{r}},\mathrm{o}_{\mathrm{p}}\right) + \theta_2\phi_{\mathrm{rp}} + \theta_3\psi_{\mathrm{rp}}\right) \quad (9)$$

where $M_{rp}$ denotes the final matching probability between candidate $r$ and position $p$ , $\sigma(\cdot)$ represents the Sigmoid mapping function, $\cos\left(\mathrm{o}_r, \mathrm{o}_p\right)$ indicates the cosine similarity between candidate and position projection vectors, $\phi_{rp}$ signifies the constraint satisfaction score, $\psi_{rp}$ measures field alignment strength, and $\theta_1, \theta_2, \theta_3$ denotes the learnable fusion weight.During training, the scoring output is co-optimized with positive and negative sample labels. In inference, it directly drives candidate set ranking. Consequently, high-scoring samples typically exhibit strong semantic alignment, high competency fulfillment, and stable field mapping relationships. Conversely, low-scoring samples-even with partial skill overlap-are ranked lower due to constraint gaps or experience mismatches.

## 6. Design of Intelligent Talent Screening and Matching Strategy

The intelligent talent screening and matching strategy requires further candidate pool compression and priority allocation beyond semantic scoring. This is because recruitment scenarios feature limited open positions, while candidate samples typically exhibit large scale, uneven quality distribution, and significant variations in capability boundaries. The screening process first constructs a joint decision metric based on semantic matching probability, job constraint fulfillment, and resume stability. This metric is then applied to perform hierarchical filtering on the candidate pool, ensuring highly relevant samples advance to the fine-tuning stage. The comprehensive screening score for candidates is defined as:

$$\mathrm{R}_{\mathrm{i}} = \xi_1\mathrm{M}_{\mathrm{i}} + \xi_2\Omega_{\mathrm{i}} + \xi_3\Lambda_{\mathrm{i}} \quad (10)$$

where $R_i$ denotes the comprehensive screening score for the $\mathrm{i}_{\mathrm{th}}$ candidate, $M_i$ represents semantic matching probability, $\Omega_i$ indicates job constraint fulfillment, $\Lambda_i$ signifies resume stability, and $\xi_1,\xi_2,\xi_3$ denotes the fusion weight. As shown in Table 3, positions with rigid skill requirements are more sensitive to $\Omega_i$ , while roles with flexible responsibilities rely more heavily on $M_i$ . Consequently, the screening strategy avoids fixed thresholds, dynamically adjusting selection thresholds based on job attributes [10]. The ranking phase subsequently performs Top-KKK candidate reordering on retained samples. Constraint gap penalties mitigate the risk of prioritizing samples with high local similarity but overall poor fit. This process directly alters the clustering of high-quality resumes in the final candidate list, manifesting in synchronized changes to accuracy, recall, and F1 scores in subsequent experiments.

**Table 3.** Key Decision Factors and Their Directional Effects in Intelligent Talent Screening.

| Decision Factor | Meaning | Direction of Influence | Impact on Screening Results |
|---|---|---|---|
| Semantic Matching Probability $M_i$ | Overall semantic alignment between candidate and position | Positive Reinforcement | The baseline probability determining a candidate's advancement to the fine-tuning stage |
| Job Constraint Satisfaction $\Omega_i$ | Degree to which hard requirements such as education, experience, and skills are met | Positive Boost | Determines whether candidates meet the position's boundary requirements |
| Resume stability $\Lambda_i$ | Continuity of employment, consistency of experience, and credibility of resume | Positive reinforcement | Determines candidate priority in cases of equal scores |
| Constraint gap penalty | Degree of deviation where semantic similarity exists but critical requirements are missing | Negative Suppression | Prevents boundary-inconsistent samples from being ranked too highly |

## 7. Experimental Design and Results Analysis

### *7.1. Experimental Dataset and Environment*

The experimental dataset requires to cover three types of information: resume text, job description text, and actual application relationships. Otherwise, the aforementioned semantic encoding results cannot be effectively validated in screening and ranking scenarios. During data construction, anonymized samples for four job categories-technical, operations, marketing, and functional support-were collected from public recruitment platforms. Fields including educational requirements, experience thresholds, skill phrases, responsibility descriptions, and candidate resumes were preserved to form a sample set suitable for joint evaluation of semantic alignment and candidate ranking. As shown in Table 4, the training, validation, and test sets maintain relative balance in job type and matching label distribution. This partitioning minimizes category bias interference in subsequent results. The experimental environment utilized a Linux server, PyTorch framework, and single-GPU training configuration. Text length is uniformly capped within a comparable range, while batch size and learning rate settings remain stable to ensure convergence of resume representations, job representations, and matching scoring under identical computational conditions. Negative sample construction follows the principle of "prioritizing confusion within similar job categories while supplementing cross-category jobs with perturbations," thereby enhancing the model's ability to identify challenging samples during subsequent generalization testing.

**Table 4.** Experimental Datasets and Environment Configuration.

| Item | Configuration Details |
|---|---|
| Data Source | Public recruitment platform anonymized samples |

| | |
|---|---|
| Job Category | Technical, Operations, Marketing, Functional Support |
| Number of Resume Samples | 18,600 |
| Number of Job Samples | 4,250 |
| Matching pairs | 52,400 |
| Data partitioning | Training set 70%, validation set 15%, test set 15% |
| Development Framework | PyTorch |
| Runtime Environment | Linux |
| Computing Device | Single-GPU Server |
| Maximum Text Length | Uniform truncation and padding |
| Negative sample strategy | Intra-class confusion + cross-class perturbation |

### *7.2. Resume-Job Matching Accuracy Testing*

The test involves inputting candidate semantic vectors and job semantic vectors into a unified test set, then performing positive/negative sample classification and accuracy statistics based on a matching probability threshold. As shown in Table 5, the constructed model achieves an accuracy of 89.47% on the test set, significantly outperforming TF-IDF + Cosine (76.28%), TextCNN (81.35%), BiLSTM (83.12%), and the single-tower BERT matching model (86.74%). The accuracy for the technical position subset is 90.63%, while operational positions achieve 88.21%.87.56% for marketing roles, and 89.02% for functional support roles. The fluctuation range across different job types was controlled within 3.07 percentage points, indicating that bilateral independent encoding and unified semantic space mapping can stably identify role boundaries and competency fulfillment relationships. Notably, technical roles showed the most significant improvement, reflecting that skill phrases, project experience, and experience thresholds possess higher discriminative power after embedding alignment.

**Table 5.** Resume-to-Position Matching Accuracy Test Results.

| Model | Overall Accuracy/% | Technical Positions/% | Operations Positions/% | Marketing Positions/% | Functional Support Positions/% |
|---|---|---|---|---|---|
| TF-IDF + Cosine | 76.28 | 77.11 | 75.42 | 74.86 | 77.03 |
| TextCNN | 81.35 | 82.47 | 80.63 | 79.94 | 82.36 |
| BiLSTM | 83.12 | 84.25 | 82.14 | 81.37 | 83.88 |
| BERT Single Tower Matching | 86.74 | 87.95 | 85.93 | 85.41 | 87.12 |
| Model constructed | 89.47 | 90.63 | 88.21 | 87.56 | 89.02 |

### *7.3. Model Recall and F1 Score Evaluation*

The test outputs statistical positive sample detection based on matching probabilities using the same test set, simultaneously calculating recall and F1 scores. Results show the constructed model achieves a recall rate of 88.63%with an F1 score of 89.04%. Both metrics surpassed the TF-IDF + Cosine model's 74.85% and 75.56%, the TextCNN model's 80.92% and 81.14%, the BiLSTM model's 82.67% and 82.89%, and the BERT single-tower matching model's 86.31% and 86.52%. In technical roles, recall reached 89.74% and F1 score 90.18%;

for operational roles, these values were 87.25% and 87.61%; for marketing roles, 86.48% and 86.97%; and for functional support roles, 88.05% and 88.43%.

### *7.4. Ablation Experiments on Different Feature Combinations*

The test sequentially removed skill semantics, project experience semantics, educational background semantics, and position constraint semantics, comparing matching performance changes on the same test set. As shown in Figure 3, the complete feature combination achieved an accuracy of 89.47%, a recall rate of 88.63%, and an F1 score of 89.04%. After removing skill semantics, the model's accuracy dropped to 85.92%, recall decreased to 84.71%, and F1 score fell to 85.28%, indicating that skill phrases remain the core discriminative source for job competency identification. After removing project experience semantics, the model achieved 86.37% accuracy, 85.46% recall, and 85.91% F1 score. This change indicates that the correspondence between project responsibilities and technical scenarios significantly impacts the assessment of candidates' actual competency. After removing educational background semantics, the model achieved 87.84% accuracy, 86.95% recall, and 87.31% F1 score-a relatively minor decline. This suggests educational information primarily serves as a boundary filter. After removing job constraint semantics, accuracy dropped to 84.58%, recall rose to 89.12%, and F1 score reached 86.79%. This outcome indicates that while weakening constraint information allows the system to recall more semantically similar samples, it simultaneously increases mis-matches, thereby loosening the job fit of top-ranked candidates.

Figure 3 Ablation experiment results for different semantic feature combinations

### *7.5. Model Generalization and Stability Analysis*

Joint evaluation of model output variability and cross-domain adaptability was conducted through cross-position category transfer, random resampling, and five rounds of independent repeated training. In the technical position training and operations position testing scenario, the model achieved an accuracy of 86.12%, recall of 85.47%, and an F1 score of 85.79%. When trained on operational roles and tested on marketing roles, the model achieved 84.95% accuracy, 84.21% recall, and an F1 score of 84.57%. When trained on mixed roles and tested on functional support roles, it achieved 88.36% accuracy, 87.74% recall, and an F1 score of 88.02%.Across five repeated experiments, the mean accuracy was 89.21% with a standard deviation of 0.43, the mean recall was 88.34% with a standard deviation of 0.51, and the mean F1 score was 88.77% with a standard deviation of 0.46.Difficult sample perturbation tests further revealed that accuracy dropped to 87.68% when 10% of key information was removed from job descriptions, while the F1 score decreased to 87.41% after randomly omitting 8% of resume fields.

## 8. Conclusions

The dual-tower matching architecture built upon Transformer-based semantic embeddings, combined with job constraint information and multidimensional semantic features to form a matching scoring and screening mechanism, demonstrated improved matching accuracy, recall, and stability in experimental results. Existing methods still rely on structured resume information and static job texts, with limited adaptability to complex resume expressions and dynamic job changes. Future research may incorporate cross-domain semantic transfer and real-time job description updating mechanisms to enhance the model's long-term generalization capabilities across diverse industry recruitment scenarios.

## References

1. Y. Liu, "Deep Learning-Based Intelligent Resume-Position Matching System: Semantic Understanding and Recommendation of BERT Model in Massive Recruitment Data," in *Proceedings of the 2025 International Symposium on Machine Learning and Social Computing*, October 2025, pp. 8-13. doi: 10.1145/3778450.3778452
2. V. Narula, R. Kumar, R. Arora, and R. Bhatia, "Enhancing job recommendations using NLP and machine learning techniques," *TIJER-International Research Journal*, pp. 826-853, 2023.

3. A. Deshmukh and A. Raut, "Applying bert-based nlp for automated resume screening and candidate ranking," *Annals of Data Science*, vol. 12, no. 2, pp. 591-603, 2025. doi: 10.1007/s40745-024-00524-5
4. C. Tuan, M. T. Dang, H. N. Do, V. K. Solanki, J. Torres, R. Gonzalez Crespo, and T. N. A. Nguyen, "Ontology and its applications in skills matching in job recruitment," *Applied Ontology*, vol. 19, no. 3, pp. 287-306, 2024.
5. F. Antony, P. Eshanth, and S. J. Sharine, "An Intelligent Resume-Job Matching System: Bridging Candidates and Opportunities through Advanced Semantic Understanding," in *2026 9th International Conference on Computational Intelligence in Data Science (ICCIDS)*, January 2026, pp. 1-6. doi: 10.1109/iccids69108.2026.11407881
6. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 10345-10425, 2023. doi: 10.1007/s10462-023-10419-1
7. Y. Wang, Y. Sun, Y. Fu, D. Zhu, and Z. Tian, "Spectrum-BERT: pretraining of deep bidirectional transformers for spectral classification of Chinese liquors," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-13, 2024. doi: 10.1109/tim.2024.3374300
8. A. Areshey and H. Mathkour, "Transfer learning for sentiment classification using bidirectional encoder representations from transformers (BERT) model," *Sensors*, vol. 23, no. 11, p. 5232, 2023. doi: 10.3390/s23115232
9. A. Choudhary and A. Arora, "Assessment of bidirectional transformer encoder model and attention based bidirectional LSTM language models for fake news detection," *Journal of Retailing and Consumer Services*, vol. 76, p. 103545, 2024. doi: 10.1016/j.jretconser.2023.103545
10. A. Portes, A. Trott, S. Havens, D. King, A. Venigalla, M. Nadeem, and J. Frankle, "MosaicBERT: A bidirectional encoder optimized for fast pretraining," *Advances in Neural Information Processing Systems*, vol. 36, pp. 3106-3130, 2023. doi: 10.52202/075280-0137