Enhancing Online Recruitment with Category-Aware MoE and LLM-based Data Augmentation

Minping Chen[1, *], Bing Xu[3, *], Zulong Chen[3, †], Chuanfei Xu[4, †], Ying Zhou[5], Zui Tao[6], Zeyi Wen[1, 2, †]
[1]HKUST (GZ) [2]HKUST [3]Alibaba Group
[4]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
[5]Zhijiang Lab [6]The Hong Kong Polytechnic University
Correspondence: zulong.czl@alibaba-inc.com, xuchuanfei@gml.ac.cn, wenzeyi@hkust-gz.edu.cn

Abstract

Person-Job Fit (PJF) is a critical component for online recruitment. Existing approaches face several challenges, particularly in handling low-quality job descriptions and similar candidate-job pairs, which impair model performance. To address these challenges, this paper proposes a large language model (LLM) based method with two novel techniques: (1) LLM-based data augmentation, which polishes and rewrites low-quality job descriptions by leveraging chain-of-thought (COT) prompts, and (2) category-aware Mixture of Experts (MoE) that assists in identifying similar candidate-job pairs. This MoE module incorporates category embeddings to dynamically assign weights to the experts and learns more distinguishable patterns for similar candidate-job pairs. We perform offline evaluations and online A/B tests on our recruitment platform. Our method relatively surpasses existing methods by 2.40% in AUC and 7.46% in GAUC, and boosts click-through conversion rate (CTCVR) by 19.4% in online tests, saving millions of CNY in external headhunting expenses.

1 Introduction

The swift advancement of Internet technology has transformed online recruitment into a widely used web service for job seekers and recruiters (Geyik et al., 2018; Kenthapadi et al., 2017; Paparrizos et al., 2011). Due to the significant increase in usage of online recruitment platforms, Person-Job Fit (PJF) (Qin et al., 2023) has emerged as an effective solution to automatically measure the matching degree between a job and a candidate.

In recent years, the PJF task has been regarded as a text matching task, by exploiting the rich semantic information in resumes and job descriptions (JDs). Various neural networks have been applied to enhance the encoding of the resumes and JDs, including CNN (Zhu et al., 2018; He et al., 2021; Zhenhong et al., 2021), RNN (Qin et al., 2018, 2020; Yan et al., 2019) and GNN (Wang et al., 2022) et al. Along with improving the text representation, existing methods also explore the candidate-job interaction. Such efforts include employing historically accepted and rejected applications (Le et al., 2019), capturing users' dynamic preferences from their multi-behavioral sequences such as click, invite/apply, chat (Yang et al., 2022) or searching histories (Hou et al., 2022), modeling the two-way interaction (Yang et al., 2022; Zheng et al., 2023), and multi-stage interaction (Zheng et al., 2024).

Despite the notable progress, existing methods still encounter these challenges: (i) Numerous low-quality job descriptions (JDs) exist in our online recruitment system (about 25% training samples), i.e., with short content and poor information. JD is a crucial input for the PJF task, and poor JDs hinder the model performance. However, this issue remains unaddressed by existing methods. ii) While existing methods have explored multiple interaction strategies, they struggle with managing similar candidate-job pairs, i.e., the job requirements and the work experience of the candidates are similar, but they differ in some key aspects. For instance, candidates for a data analyst position may share relevant experience with algorithm engineers, e.g., basic data processing and algorithm development. However, algorithm engineers primarily design and optimize algorithms for applications, while data analysts interpret data to provide insights for business decisions. Therefore, a data analyst candidate is not a suitable match for an algorithm engineer position.

To address these challenges, we propose a large language model (LLM) based method with data augmentation and mixture of experts (MoE) (Jacobs et al., 1991; Shazeer et al., 2017; Lepikhin et al., 2020) architecture. Specifically, we introduce an LLM-based data augmentation module to polish

[*] Equal contribution.
[†] Corresponding author.

or rewrite the low-quality JDs. Chain-of-thought (COT) (Wei et al., 2022) prompt templates are carefully designed to instruct the LLM to complete this task following the given requirements, with several resumes that have passed the interviews as references. Furthermore, to improve the model's ability to identify similar candidate-job pairs, we propose a category-aware MoE module that leverages the category information of the jobs and candidates. This category information is input to the gating net to dynamically assign suitable weights for each expert. Each expert, acting like a human expert, is responsible for processing a different scenario and capturing the fine-grained distinction of similar texts and historical interactions. We summarize the contributions of this paper as follows.

1. We introduce an LLM-based data augmentation strategy to refine low-quality job descriptions, a challenge that remains unaddressed in existing methods, thereby improving the overall model performance.

2. We propose a novel category-aware MoE module to enhance the matching degree assessment. With this module, our method can learn more distinguishable patterns for similar candidate-job pairs.

3. We conduct offline experiments and online A/B tests on our recruitment platform. Our method relatively outperforms the state-of-the-art method by 2.40% in AUC and 7.46% in GAUC, and boosts the click-through conversion rate (CTCVR) in online tests by 19.4%.

2 Methodology

2.1 Notation and Problem Statement

Let $C = \{c_1, c_2, ..., c_m\}$ be the candidates, and $J = \{j_1, j_2, ..., j_n\}$ be the jobs, with m and n as their sizes. Each candidate $c \in C$ has a resume text r and a category t^{c}. Each job $j \in J$ has a description text d and a category t^{j}. Candidates have historical job interaction sequences: jobs that the resume of the candidate has been evaluated $\mathcal{J}^{\mathrm{e}} = \{j_1, j_2, ...j_{k_1}\}$, jobs passed resume evaluation $\mathcal{J}^{\mathrm{pe}} = \{j_1, j_2, ...j_{k_2}\}$, and jobs passed the interviews $\mathcal{J}^{\mathrm{pi}} = \{j_1, j_2, ...j_{k_3}\}$. Similarly, jobs have candidate interaction sequences: evaluated candidates $\mathcal{C}^{\mathrm{e}} = \{c_1, c_2, ...c_{k_4}\}$, candidates passed resume evaluation $\mathcal{C}^{\mathrm{pe}} = \{c_1, c_2, ...c_{k_5}\}$, and candidates passed the interviews $\mathcal{C}^{\mathrm{pi}} = \{c_1, c_2, ...c_{k_6}\}$. The numbers $\{k_1, k_2, ...k_6\}$ represent the sequence sizes. Given these inputs, our model predicts if candidate c_i matches job j_i with binary output y_i.

2.2 Overview of Our Method

We present an overview of our method, as shown in Figure 1. It comprises three modules: LLM-based job description augmentation module, fine-grained historical interaction module and category-aware MoE module. The model processes the candidate information and job information, converting them into embeddings. To tackle the issue of low-quality job descriptions (JDs), we introduce a data augmentation strategy based on LLMs. Then the fine-grained historical interaction module learns the bilateral and comprehensive interactions between the candidates and jobs, aligning with the recruitment process to enhance effectiveness. The category-aware MoE module further refines the interaction encoding by incorporating category information to address the challenge of similar candidate-job pairs, ultimately producing predictions.

2.3 LLM-based JD Augmentation

Our online recruitment system has many short, low-quality job descriptions (25% of training samples). To address this, we leverage an LLM to enhance JDs below a certain character-level length threshold l, as updating all JDs may introduce noise. We conduct multiple rounds of optimization on a JD subset to enhance the COT prompt template. The prompt is based on the original JD and its historical resumes from candidates who passed the interviews, offering references for the LLM to refine the JDs. To tackle potential LLM hallucination issues, we carefully design the prompt, as illustrated in Figure 2. The **system instruction** guides the LLM on its role and introduces the task. The LLM needs to act as an "HR expert" with a professional background in human resources of various industries, and tasks like resume analysis and job matching.

In the **requirement part**, we provide clear task descriptions, which is divided into multiple steps based on the COT technique. First, the LLM evaluates the completeness of the original JD, then extracts the key information, e.g., job position and professional skills. For JDs that lack historically matched resumes, the LLM leverages its comprehensive HR expertise to enhance them. Otherwise, the LLM summarizes the common industry areas and professional requirements in multiple resumes, and integrates the results into the original JD to

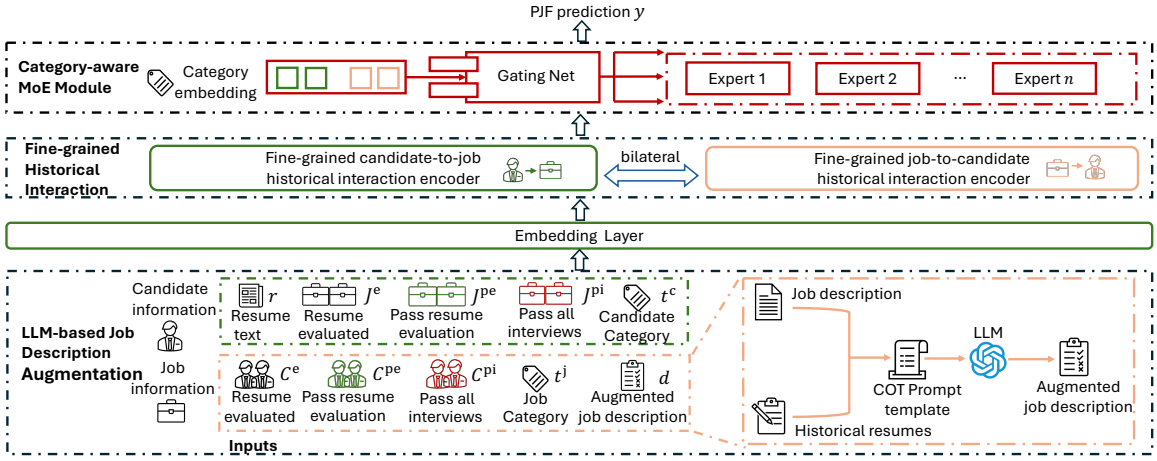


Figure 1: The overview of our method.

improve it. To ensure quality, we require the LLM to retain over 70% of the original keywords, avoid complexity or vagueness, and maintain clear, concise professionalism. These guidelines help produce effective JDs and prevent excessive rewriting noise. The full prompt is presented in Appendix A.

2.4 Fine-grained Historical Interaction

The fine-grained historical interaction module uses two identical encoders to learn the bilateral interaction between candidates and jobs. The only difference between the encoders is their input. We present one of them in Figure 3. For simplicity, we use $\{r_i, d_i, \mathcal{J}_i^{\mathrm{e}}, \mathcal{J}_i^{\mathrm{pe}}, \mathcal{J}_i^{\mathrm{pi}}, \mathcal{C}_i^{\mathrm{e}}, \mathcal{C}_i^{\mathrm{pe}}, \mathcal{C}_i^{\mathrm{pi}}\}$ to represent the input embeddings processed by the interaction encoder. Our bilateral interaction is more fine-grained and comprehensive than existing methods. We break down interaction modeling into procedures reflecting the recruitment process: finishing resume evaluation, passing resume evaluation and passing all interviews. For each procedure, we both learn internal and external interactions. Internal interaction involves the JD embedding d_i and its historical interacted candidate sequences $\mathcal{C}_i^{\mathrm{e}}/\mathcal{C}_i^{\mathrm{pe}}/\mathcal{C}_i^{\mathrm{pi}}$ (or the resume embedding r_i and its historical interacted job sequences $\mathcal{J}_i^{\mathrm{e}}/\mathcal{J}_i^{\mathrm{pe}}/\mathcal{J}_i^{\mathrm{pi}}$), and external interaction involves d_i and $\mathcal{J}_i^{\mathrm{e}}/\mathcal{J}_i^{\mathrm{pe}}/\mathcal{J}_i^{\mathrm{pi}}$ (or r_i and $\mathcal{C}_i^{\mathrm{e}}/\mathcal{C}_i^{\mathrm{pe}}/\mathcal{C}_i^{\mathrm{pi}}$). We utilize the multi-head attention (Vaswani, 2017) to learn these interactions, and take the internal candidate-to-job interaction of finishing resume evaluation as an example:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_{\mathrm{k}}}})V,$$

$$\mathrm{MultiHead}(r_i, \mathcal{J}_i^{\mathrm{e}}) = \mathrm{Concat}(head_i, head_2, ...head_h)W^{\mathrm{O}},$$

$$\text{where } head_i = \mathrm{Attention}(r_i W_i^{\mathrm{Q}}, \mathcal{J}_i^{\mathrm{e}} W_i^{\mathrm{K}}, \mathcal{J}_i^{\mathrm{e}} W_i^{\mathrm{V}})$$

Here $W_i^{\mathrm{Q}} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{k}}}$, $W_i^{\mathrm{K}} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{k}}}$, $W_i^{\mathrm{V}} \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{v}}}$, $W^{\mathrm{O}} \in \mathbb{R}^{hd_{\mathrm{v}} \times d_{\mathrm{model}}}$ are the weight matrices, d_{model} is dimension of the embedding, h is the number of attention heads and $d_{\mathrm{k}} = d_{\mathrm{v}} = d_{\mathrm{model}}/h$. For the external interaction in this example, we only need to replace Q with $d_i W_i^{\mathrm{Q}}$. The learned bilateral interactions are concatenated into a DNN for feature fusion.

2.5 Category-aware MoE Module

Initially, a category vocabulary is defined by the HRs in our company based on industry standards, and the reliability of category assignment for the jobs and candidates is guaranteed through manual verification. Subsequently, we learn an embedding matrix $E_{\mathrm{c}} \in \mathbb{R}^{n_{\mathrm{c}} \times d_{\mathrm{e}}}$ for this vocabulary, where n_{c} is the vocabulary size and d_{e} is the embedding dimension. E_{c} is randomly initialized and optimized during training. Each expert in the category-aware MoE module is a feed-forward network (FFN), aiming to learn distinct candidate-job matching patterns across various job categories. Like a human expert, each expert specializes in different scenarios that cover multiple job categories. Furthermore, they capture the distinction and learn a feature fusion of the bilateral and fine-grained candidate-job interactions. In other words, the MoE module leverages not only the category information but also the representation and interaction distinctions to identify similar candidate-job pairs. The gating net is a two-layer FFN, guided by the category embedding e_{c} to dynamically assign appropriate weights to each expert. The final prediction is based on the weighted sum of the output of each expert:

$$G_i = \mathrm{softmax}(W_2^{\mathrm{G}}(\mathrm{ReLU}(W_1^{\mathrm{G}} e_{\mathrm{c}} + b_1^{\mathrm{G}})) + b_2^{\mathrm{G}}),$$

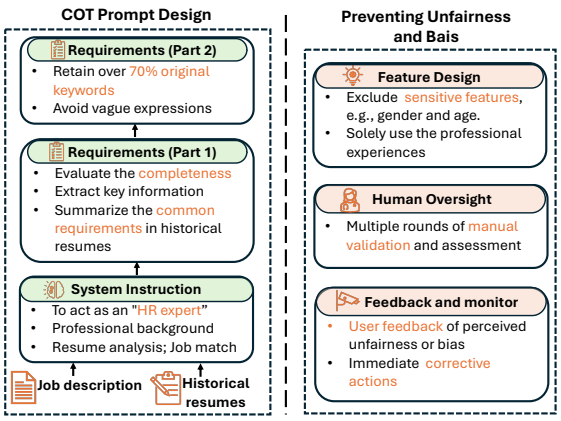


Figure 2: Prompt and strategies design in our method.

$$E_i(x) = W_3^{\mathrm{E}}(\mathrm{ReLU}(W_2^{\mathrm{E}}(\mathrm{ReLU}(W_1^{\mathrm{E}}x + b_1^{\mathrm{E}}) + b_2^{\mathrm{E}})) + b_3^{\mathrm{E}},$$

$$y = \sum_{i=1}^{n_{\mathrm{e}}} G_i E_i(x),$$

where $E_i(x)$ is the output of the i-th expert, G_i is the gating weight of the i-th expert, x is the joint representation of the input, n_{e} is the number of experts, W_i^{G} and W_i^{E} are the weights of the gating network and expert, and y is the final prediction.

We train our model in a pair-wise way using the BPR loss (Rendle et al., 2012) to optimize the ranking between positive and negative samples, adding a regularization item to avoid overfitting:

$$\mathcal{L} = -\frac{1}{|B|} \sum_{(y_i^+, y_i^-) \in B} \log(\sigma(y_i^+ - y_i^-)) + \lambda(\frac{1}{|B|} \sum_{(y_i^+, y_i^-) \in B} (y_i^+)^2 + (y_i^-)^2),$$

where B is a batch of training data, y_i^+ and y_i^- are the prediction scores of the positive and negative samples, respectively, σ denotes the sigmoid function, and λ is the regularization weight.

2.6 Strategies for Preventing Bias

To ensure that the AI model in our system does not introduce unfairness or biases to the decision-making, we develop several strategies in Figure 2:

Feature design for fairness: Our model intentionally excludes sensitive features that may lead to unfairness and biases as inputs, e.g., gender and age. However, neural networks are susceptible to bias arising from proxy features, such as educational institutions, graduation years, and geographic location. To address this, we do not use such proxy features as explicit inputs. We solely use the professional experiences of the talents for PJF modeling, ensuring that the model focuses on evaluating professional qualifications, skills, and experiences rather than features having a risk of introducing unfairness and biases.

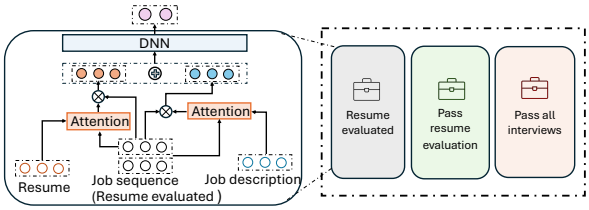


Figure 3: The architecture of the fine-grained candidate-to-job historical interaction encoder.

Human Oversight and Feedback: It is important to note that our method is used as a decision support tool rather than an automated filter. Candidates are ranked at the user interface based on model predictions, and recruiters retain full control over hiring decisions. Our recruitment process is fortified by rigorous human oversight, with multiple rounds of manual validation and assessment to ensure fairness. Additionally, it is equipped with a feedback and monitoring mechanism, empowering users to report instances of perceived bias. This facilitates immediate corrective actions and reinforces our commitment to fairness.

3 Experiments

3.1 Data and Baselines

Data Collection In the PJF problem, the only one open-source dataset, i.e., Aliyun dataset (which has 36274 training samples, 300 validation samples, and 300 test samples)[1] is not suitable for our scenario, as it misses some key information (e.g., job categories and historical interactions). Thus, we collect a real-world dataset from our online recruitment system, creating training data by randomly sampling matched candidate-job pairs for positive samples and unmatched candidates for negative samples. The training data contains about 8 million samples (47K jobs and 0.8 million resumes). The test data is also collected from our system, consisting of 25K samples (2K jobs and 20K resumes). Note that our data split is strictly temporal: training data is collected from an earlier period (up to mid-July 2024), while test data is from a later period (from mid to end of July 2024), ensuring no future information leaks into training. Additionally, we perform candidate and job deduplication based on their unique IDs. Our code is available here[2].

[1] https://tianchi.aliyun.com/competition/entrance/231728
[2] https://github.com/Chan-1996/LLM-PJF

Table 1: Performance comparison on our dataset.

	Method	AUC	GAUC	NDCG	AP
Traditional ML	LR	0.672	0.615	0.827	0.235
	XGBoost	0.673	0.616	0.830	0.237
Similarity based	DSSM	0.653	0.609	0.819	0.216
	BGE-Raw	0.606	0.592	0.803	0.187
PJF model	PJFNN	0.635	0.634	0.811	0.200
	IPJF	0.669	0.630	0.832	0.237
	PJFFF	0.650	0.643	0.804	0.197
	SHPJF	0.651	0.630	0.810	0.204
	CONFIT	0.707	0.661	**0.842**	0.260
	Ours	**0.724**	**0.709**	0.837	**0.262**

Evaluation Metrics For offline evaluations, we use AUC, grouped AUC (GAUC), normalized discounted cumulative gain (NDCG), and average precision score (AP) as evaluation metrics. AUC and GAUC are not fully linearly correlated because GAUC groups samples by jobs, and higher GAUC is achieved only when samples within the same job are ranked more accurately. For online A/B tests, we use click-through conversion rate (CTCVR) (Ma et al., 2018) as the evaluation metric. A higher CTCVR indicates recruiters find suitable candidates more accurately and with less effort, reducing the number of resumes to review. Detailed definitions of these metrics are in Appendix C.

Baselines We compare our method with various baselines: Logistic Regression (LR), XGBoost (Chen and Guestrin, 2016), DSSM (Huang et al., 2013), BGE-Raw (Chen et al., 2024), PJFNN (Zhu et al., 2018), IPJF (Le et al., 2019), PJFFF (Jiang et al., 2020), SHPJF (Hou et al., 2022), and CONFIT (Yu et al., 2024). Introduction of these baselines and other implementation details are presented in Appendix C.

3.2 Main Results

Offline Evaluation The offline comparison is presented in Table 1. First, LR and XGBoost can effectively rank obvious positive samples, and get high NDCG due to their sensitivity to distinct features like exact matches. They also achieve better performance than the similarity-based deep learning methods. This indicates that cosine similarity cannot well distinguish different matching degrees in the PJF task. Compared with CONFIT, our method achieves a relative improvement of 2.40% and 7.46% in AUC and GAUC, respectively. Although the NDCG of our method drops, the degradation is negligible (i.e., only -0.59%). This is because our solution works better for matching than ranking (higher AUC), while CONFIT is a recall model, focusing on ranking (slight higher NDCG). However, we achieve better overall performance.

Table 2: Performance comparison on Aliyun dataset.

Method	AUC	GAUC	NDCG	AP
CONFIT	0.523	0.494	0.814	0.494
Ours	**0.558**	**0.565**	**0.884**	**0.674**

We also evaluate on the Aliyun dataset to verify the effectiveness of our method when key information like job categories and historical interactions is not available. As shown in Table 2, our method consistently outperforms CONFIT in all metrics. This suggests that better feature modeling also contributes to our performance improvement.

Online Evaluation We perform online A/B tests for seven working days on our recruitment platform, strictly following the principles of Google, e.g., randomization and single key metric. The results are shown in Figure 4. The online baseline is PJFFF, while CONFIT isn't compared due to its high overall latency (500ms vs. the required <300ms). For fair comparison, the job seekers and recruiters for the two methods were randomly selected in a 1:1 ratio. Additionally, our A/A test shows no significant difference between the groups, i.e., p-value > 0.05. Our method outperforms the baseline for all days. Due to confidentiality, we do not disclose the actual CTCVR values, but our method shows an average relative improvement of 19.38%, statistically significant with a p-value of $0.0035 < 0.01$. Beyond CTCVR, we also monitor daily AUC metric, real-time user feedback through likes and dislikes, collection of bad cases, and regular user surveys. Moreover, our method helps save millions of CNY in headhunting expenses, as nearly 40% of hires come from our internal talent database, reducing dependence on external channels. In terms of system efficiency, the LLM data augmentation is processed offline to ensure the overall latency remains below 300ms, and our model can perform inference even on CPUs since it is only about 0.5B. Besides, our system updates in real-time with a T+1 mode, where JDs are updated the following day, allowing new JDs with emerging

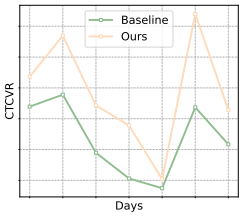


Figure 4: A/B tests.

Table 3: Ablation study.

Method	AUC	GAUC
Ours (full model)	**0.724**	**0.709**
w/o JD Aug.	0.718	0.702
w/o MoE	0.679	0.665
w/o Int.	0.717	0.695
w/o JD Aug. & MoE	0.675	0.664
w/o JD Aug. & MoE & Int.	0.664	0.648

skills to enter the pipeline.

3.3 Discussion

Ablation Study We disable each module in our method separately to verify the effectiveness of them, as shown in Table 3. *w/o JD Aug.* and *w/o MoE* refer to the model without using the LLM-based JD augmentation and the category-aware MoE module, respectively. *w/o Int.* excludes the fine-grained historical interaction encoding, using only the interaction of passing the resume evaluation. Disabling any module or their combinations leads to performance drops, confirming their effectiveness for the PJF task. The most significant performance drop occurs when the category-aware MoE module is removed, highlighting its crucial role in enhancing the model's ability to differentiate similar candidate-job pairs.

Impact of the JD Augmentation To evaluate how the length threshold l affects performance, we evaluate our method with various l, as shown in the first seven rows of Table 4. Methods using JD augmentation consistently outperform the one without it, labeled *Ours-w/o JD Aug.*, proving its effectiveness. Moreover, l has small effect on performance, highlighting the robustness of our strategy. To assess whether the data augmentation enhances model performance on samples with low-quality JDs, we calculate the AUC and GAUC for these test samples, which comprise 9% of the test set. The last three rows in Table 4 show that the model with JD augmentation significantly outperforms the one without by 3.4% in AUC and 7.0% in GAUC. We also test one of our baselines, PJFFF, when provided with our augmented JDs, observing notable improvements over the vanilla PJFFF. This further supports the effectiveness of our LLM-based JD augmentation strategy.

Impact of the LLM Hallucination Issue As discussed in Section 2.3, we carefully design the prompt template to reduce LLM hallucination. To assess its effectiveness, we randomly sample fifty LLM-augmented JDs covering eight job categories, and manually check them for hallucination issues. The results are shown in Table 5. Only one JD suffers from the hallucination issue, and the ratio is 2%, indicating the minor influence of the hallucination issue. Moreover, our system incorporates human oversight as introduced in Section 2.6 to further minimize the effects of hallucination issues.

Table 4: Impact of the job description augmentation.

Threshold l	AUC (all)	GAUC (all)
$l = 200$	0.724	**0.709**
$l = 300$	0.726	0.705
$l = 350$	0.726	0.706
$l = 400$	0.726	0.706
w/o using l	0.724	0.704
Ours-w/o JD Aug.	0.718	0.702
Ours	**0.724**	**0.709**
PJFFF	0.650	0.643
PJFFF + JD Aug.	**0.665**	**0.652**
Method	AUC (JD len < 200)	GAUC (JD len < 200)
Ours-w/o JD Aug.	0.698	0.512
Ours	**0.732**	**0.582**

Table 5: Statistics on hallucination issues.

#No Hallucination	#Minor[1]	#Moderate[2]	#Severe[3]
49	1	0	0

[1]Minor: Slight variations. [2]Moderate: Noticeable differences. [3]Severe: Significant deviation.

Expert Weights Visualization Figure 5 shows the expert weights for various job categories. Specifically, we analyze the test set, calculating the average expert weight for each job category. Contributions of the five experts vary significantly across different person-job fitting scenarios, each encompassing several job categories. This highlights our method's ability to adaptively integrate the judgments of multiple experts tailored to specific scenar-

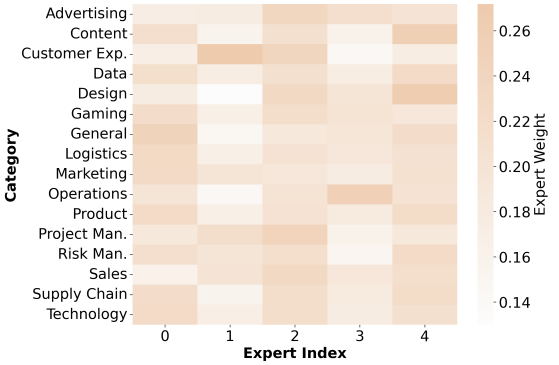


Figure 5: Visualization of the expert weights.

ios. Furthermore, each expert assesses candidate-job pairs from different perspectives, akin to the diverse evaluations by human professionals.

4 Related Work

In recent years, the PJF task has been regarded as a text matching task, aiming to fully utilize the rich semantic information in resumes and job descriptions. Various neural networks, such as CNN (Zhu et al., 2018; Maheshwary and Misra, 2018; He et al., 2021; Zhenhong et al., 2021), RNN (Qin et al., 2018, 2020; Yan et al., 2019) and GNN (Wang et al., 2022) have been employed to encode resumes and JDs. However, these methods frequently overlook candidate-job interactions.

LLM-based methods CONFIT (Yu et al., 2024) uses LLMs to increase training samples by paraphrasing specific sections in resumes or job posts. In contrast, our method enhances low-quality job descriptions by polishing and rewriting based on historically matched resumes, which is orthogonal to CONFIT. Another method LGIR (Du et al., 2024) improves job recommendations by combining explicit and implicit user data for better resume completion using LLMs. MockLLM (Sun et al., 2024) introduces a mock interview and two-sided evaluation method, using an LLM agent to simulate both interviewer and candidate roles. Our method is also orthogonal to them, as LGIR focuses on resume completion, while MockLLM uses LLMs for mock interviews. A detailed literature review is provided in Appendix D.

5 Conclusion

This paper presents innovative solutions to the persistent challenges in the Person-Job Fit (PJF) task, specifically addressing the issues of low-quality job descriptions and similar candidate-job pairs. By introducing an LLM-based method based on a data augmentation strategy, alongside the fine-grained historical interaction module and the category-aware MoE module, our approach significantly enhances the person-job matching process. Results from both offline experiments and online A/B tests demonstrate the superiority of our method.

Limitations

Our system depends on the availability of historical interaction sequences, which may be sparse for new users or job posts, potentially affecting matching accuracy in cold-start scenarios. In practice, we can leverage LLM-generated pseudo-histories to improve robustness when interaction data is sparse. We plan to address this issue by developing a domain-specific LLM that can provide pre-assessments for the candidates and simulate the interview process to accumulate historical data. Additionally, our LLM-based data augmentation strategy relies on a fixed-length threshold to identify low-quality job descriptions, which may not fully capture the nuanced variability in content richness across different industries or job categories.

Ethical considerations

In conducting our research, we have prioritized ethical considerations to ensure the responsible use of data throughout the study. Our collected real-world data mainly consists of job descriptions and resumes of talents. To protect the privacy of users, we only use the professional experiences in their resumes as inputs for our person-job fit model and remove personally identifiable information. We also mitigate any potential biases in the dataset. As introduced in Section 2.6, sensitive features that may lead to unfairness and biases, e.g., gender and age, are excluded. Additionally, our system is equipped with human oversight and feedback mechanisms, ensuring that any potential bias can be promptly addressed.

Acknowledgments

This work was supported by the Key R&D Program of Zhejiang (2024C01036). It was also funded by the NSFC Project (No. 62306256) and the Natural Science Foundation of Guangdong Province (No. 2025A1515010261).

References

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD 2016*, pages 785–794.

Mamadou Diaby, Emmanuel Viennet, and Tristan Launay. 2013. Toward the next generation of recruitment tools: an online social network-based job recommender system. In *Proceedings of the 2013*

IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pages 821–828.

Yingpeng Du, Di Luo, Rui Yan, Xiaopei Wang, Hongzhi Liu, Hengshu Zhu, Yang Song, and Jie Zhang. 2024. Enhancing job recommendation through llm-based generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2024*, volume 38, pages 8363–8371.

Bin Fu, Hongzhi Liu, Yao Zhu, Yang Song, Tao Zhang, and Zhonghai Wu. 2021. Beyond matching: Modeling two-sided multi-behavioral sequences for dynamic person-job fit. In *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021*, pages 359–375.

Sahin Cem Geyik, Qi Guo, Bo Hu, Cagri Ozcaglar, Ketan Thakkar, Xianren Wu, and Krishnaram Kenthapadi. 2018. Talent search and recommendation systems at linkedin: Practical challenges and lessons learned. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*, pages 1353–1354.

Miao He, Tao Wang, Yuanyuan Zhu, Yingguo Chen, Feng Yao, and Ning Wang. 2021. Finn: Feature interaction neural network for person-job fit. In *The 7th International Conference on Big Data and Information Analytics, BigDIA 2021*, pages 123–130.

Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Shuqing Bian, Yang Song, Tao Zhang, and Ji-Rong Wen. 2022. Leveraging search history for improving person-job fit. In *International Conference on Database Systems for Advanced Applications, DASFAA 2022*, pages 38–54.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using click-through data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM 2013*, pages 2333–2338.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. 2020. Learning effective representations for person-job fit by feature fusion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM 2020*, pages 2549–2556.

Krishnaram Kenthapadi, Benjamin Le, and Ganesh Venkataraman. 2017. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017*, pages 346–347.

Ran Le, Wenpeng Hu, Yang Song, Tao Zhang, Dongyan Zhao, and Rui Yan. 2019. Towards effective and interpretable person-job fitting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*, pages 1883–1892.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Yao Lu, Sandy El Helou, and Denis Gillet. 2013. A recommender system for job seeking and recruiting website. In *Proceedings of the 22nd International Conference on World Wide Web, WWW 2013*, pages 963–966.

Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*, pages 1137–1140.

Saket Maheshwary and Hemant Misra. 2018. Matching resumes to jobs via deep siamese network. In *Companion Proceedings of the The Web Conference 2018, WWW 2018*, pages 87–88.

Jochen Malinowski, Tobias Keim, Oliver Wendt, and Tim Weitzel. 2006. Matching people and jobs: A bilateral recommendation approach. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 6, pages 137c–137c. IEEE.

Ioannis Paparrizos, B Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems, RecSys 2011*, pages 325–328.

Chuan Qin, Le Zhang, Yihang Cheng, Rui Zha, Dazhong Shen, Qi Zhang, Xi Chen, Ying Sun, Chen Zhu, Hengshu Zhu, and 1 others. 2023. A comprehensive survey of artificial intelligence techniques for talent analytics. *arXiv preprint arXiv:2307.03195*.

Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. 2018. Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In *The 41st international ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*, pages 25–34.

Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. 2020. An enhanced neural network approach to person-job fit in talent recruitment. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–33.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Yi Su, Magd Bayoumi, and Thorsten Joachims. 2022. Optimizing rankings for recommendation in matching markets. In *Proceedings of the ACM Web Conference 2022, WWW 2022*, pages 328–338.

Hongda Sun, Hongzhan Lin, Haiyu Yan, Chen Zhu, Yang Song, Xin Gao, Shuo Shang, and Rui Yan. 2024. Facilitating multi-role and multi-behavior collaboration of large language models for online job seeking and recruiting. *arXiv preprint arXiv:2405.18113*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, NeurIPS 2017*.

Ziyang Wang, Wei Wei, Chenwei Xu, Jun Xu, and Xian-Ling Mao. 2022. Person-job fit estimation from candidate profile and related recruitment history with co-attention neural networks. *Neurocomputing*, 501:14–24.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems, NeurIPS 2022,*, 35:24824–24837.

Rui Yan, Ran Le, Yang Song, Tao Zhang, Xiangliang Zhang, and Dongyan Zhao. 2019. Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, SIGKDD 2019*, pages 914–922.

Chen Yang, Yupeng Hou, Yang Song, Tao Zhang, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Modeling two-way selection preference for person-job fit. In *Proceedings of the 16th ACM Conference on Recommender Systems, RecSys 2022.*, pages 102–112.

Kaichun Yao, Jingshuai Zhang, Chuan Qin, Peng Wang, Hengshu Zhu, and Hui Xiong. 2022. Knowledge enhanced person-job fit for talent recruitment. In *2022 IEEE 38th International Conference on Data Engineering, ICDE 2022*, pages 3467–3480.

Xiao Yu, Jinzhong Zhang, and Zhou Yu. 2024. Confit: Improving resume-job matching using data augmentation and contrastive learning. In *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024*, pages 601–611.

Yingya Zhang, Cheng Yang, and Zhixiang Niu. 2014. A research of job recommendation system based on collaborative filtering. In *The Seventh International Symposium on Computational Intelligence and Design*, volume 1, pages 533–538. IEEE.

Bowen Zheng, Yupeng Hou, Wayne Xin Zhao, Yang Song, and Hengshu Zhu. 2023. Reciprocal sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023*, pages 89–100.

Zhi Zheng, Xiao Hu, Shanshan Gao, Hengshu Zhu, and Hui Xiong. 2024. Mirror: A multi-view reciprocal recommender system for online recruitment. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024*, pages 543–552.

Jiang Zhenhong, Peng Lingxi, and Shi Lei. 2021. Person-job fit model based on sentence-level representation and theme-word graph. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, volume 5, pages 1902–1909. IEEE.

Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. *ACM Transactions on Management Information Systems (TMIS)*, 9(3):1–17.

A COT Prompt Template

An example of the complete prompt utilized in our LLM-based data augmentation is illustrated in Figure 6 (Part 1) and Figure 7 (Part 2). Key elements within the prompt are emphasized to draw attention to their significance. We customize the prompt to adapt to different job categories for better data augmentation quality.

B System Workflow

Here, we present a comprehensive overview of the workflow of our online system. As illustrated in Figure 8, our system is architected with two primary components: the online serving module and the offline training module. The online serving module performs essential pre-processing, including parsing the real-time job description (JD) provided by the user, recalling and pre-ranking talents via a search engine, and constructing features for both the JDs and talents. These features are subsequently input to the PJF service system to compute the ranking scores for the candidates. The offline training module is responsible for optimizing the PJF model, which is enhanced with the LLM-based JD augmentation module by leveraging the historical resumes. The refined PJF model, along with the augmented JDs, is then uploaded to the PJF service system for online inference.

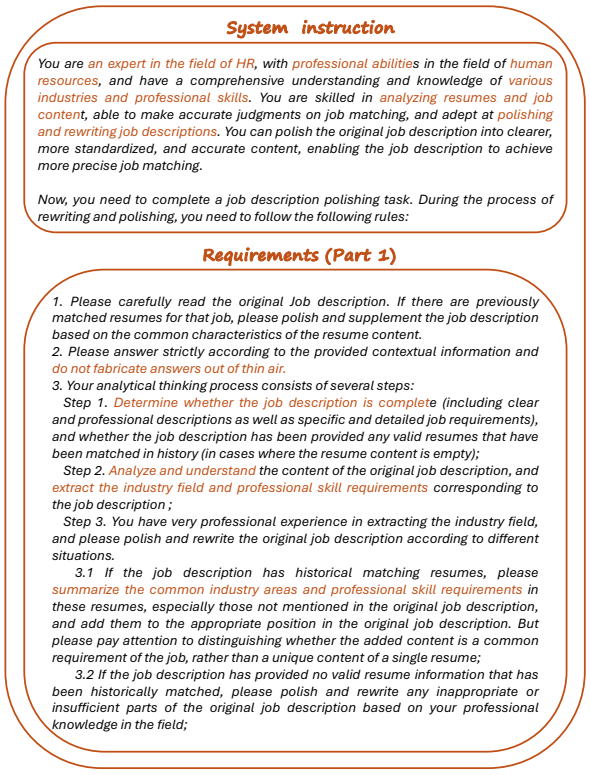


Figure 6: An example of the designed prompt (Part 1).

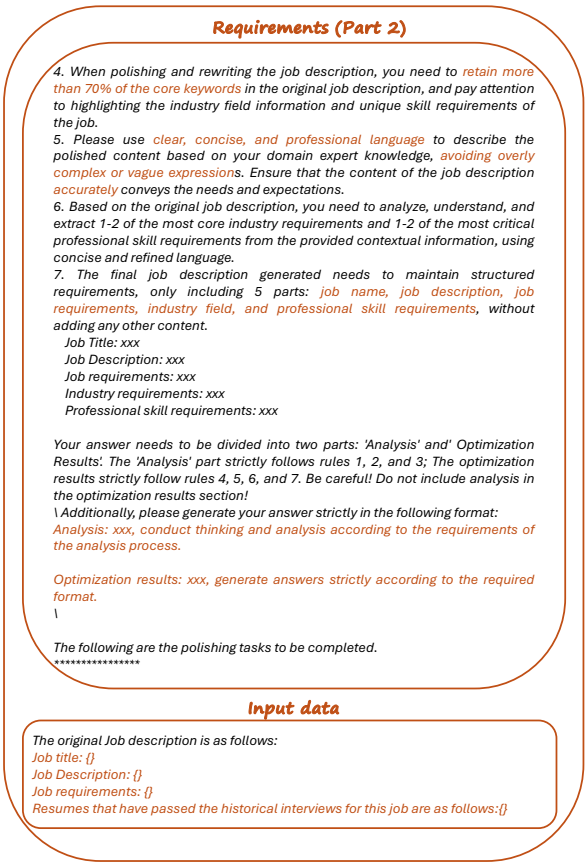


Figure 7: An example of the designed prompt (Part 2).

C Experimental Setup

This section introduces the key implementation and baseline details. We divide a small subset from the training data for hyperparameter tuning, and the test set is only used for final evaluation.

Embedding We use the fine-tuned BGE-M3 model (Chen et al., 2024) to convert the input texts, i.e., resumes and job descriptions, into 1024-dimensional embeddings. The category vocabulary size n_{c} is 16, and the category embedding dimension d_{e} is 8. Specifically, we have the following categories: Technology, Product, Supply Chain, Logistics, Content, Customer Experience, Marketing, Advertising, Data, Gaming, General, Design, Operations, Sales, Project management, and Risk management. The text embedding matrix is frozen, while the category embedding matrix is trainable.

LLM for Data Augmentation We use Qwen1.5-72B-Chat[3] to perform data augmentation for the low-quality job descriptions. We find that the quality of augmented job descriptions generated by Qwen1.5-72B-Chat is comparable to those produced by Qwen2.5-72B and Qwen3, with no significant differences observed. Our experimental results also demonstrate that Qwen1.5-72B-Chat is strong enough for job description augmentation.

Model Configuration The character-level length threshold l for LLM-based job description augmentation is 200. The category-aware MoE module has five experts, and the historical interaction sequence length is set to 20, with a padding operation for those historical interactions shorter than 20. In the multi-head attention, d_{model} is 1024 and h is 2.

Details of Evaluation Metrics The definitions of GAUC, normalized discounted cumulative gain (NDCG), average precision (AP), and click-through conversion rate (CTCVR) are as follows:

(1) GAUC:

$$GAUC = \sum_{i=1}^{N_{\mathrm{J}}} \frac{N_i^{\mathrm{J}}}{N_{\mathrm{t}}} AUC_i,$$

where N_{J} is the number of jobs in the test set, N_i^{J} is the number of test samples associated with the i-th job, and N_{t} is the total number of test samples.

(2) NDCG:

$$DCG = \sum_{i=1}^{N_{\mathrm{t}}} \frac{\hat{y}_i}{log_2(i+1)},$$

[3]https://qwenlm.github.io/blog/qwen1.5/

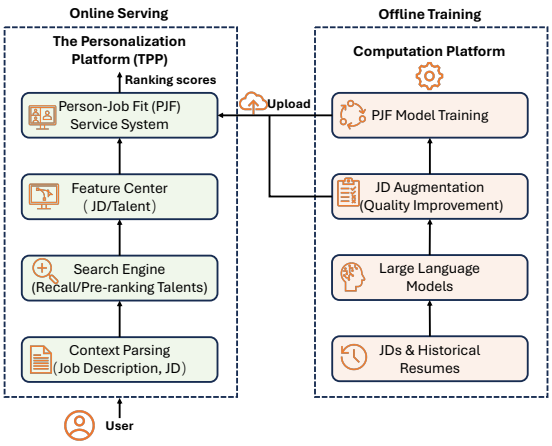


Figure 8: Workflow of our online system.

$$IDCG = \sum_{i=1}^{N_t} \frac{\check{y}_i}{log_2(i+1)},$$

$$NDCG = \frac{DCG}{IDCG},$$

where $\hat{y}_i$ and $\check{y}_i$ are the ground truths of the i-th test sample, where the test samples are sorted in descending order according to the corresponding predicted scores and the ground truths, respectively.

(3) AP:

$$AP = \sum_{i=0}^{N_{\text{th}}-1} (recall_i - recall_{i+1}) \times precision_i,$$

where N_{th} is the number of thresholds to transform a prediction score into a binary class, $recall_i$ and $precision_i$ are the recall and precision on the test set at the i-th threshold.

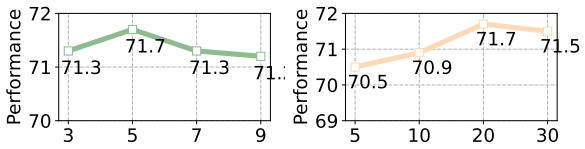
(4) CTCVR:

$$CTR = \frac{N_{\text{click}}}{N_{\text{PV}}}, CVR = \frac{N_{\text{application}}}{N_{\text{click}}}$$

$$CTCVR = CTR * CVR = \frac{N_{\text{application}}}{N_{\text{PV}}},$$

where N_{click} is the total number of clicks, N_{PV} is the total search exposure, and $N_{\text{application}}$ is the number of application initiations in our system. A higher CTCVR value indicates a more streamlined process for recruiters, enabling them to locate ideal candidates with greater accuracy and reduced effort, thereby minimizing the number of resumes they need to review.

Training Setup The training batch size, learning rate, and the weight of the regularization item are set to 256, 1e-4, and 0.1, respectively. Given our large training dataset (about 8 million), the model is sufficiently trained after one epoch, so we evaluate the test set after one epoch.



(a) Number of experts (b) Interaction sequence length

Figure 9: Impact of two hyperparameters on the final performance (average result of AUC and GAUC).

Baseline Details We perform hyperparameter tuning for all baselines to obtain their optimal performance on our dataset. Here, we provide a brief introduction of each baseline in our experiments:

1. **Logistic Regression (LR)/XGBoost** (Chen and Guestrin, 2016): The text embeddings in our method are used as features for prediction.
2. **DSSM** (Huang et al., 2013): It projects the resumes and the job descriptions into a common low-dimensional space and uses cosine similarity to determine their matching degree.
3. **BGE-Raw** (Chen et al., 2024): It uses the BGE-M3 model to produce the embeddings of the resumes and the job descriptions, with matching determined by cosine similarity.
4. **PJFNN** (Zhu et al., 2018): It encodes the resumes and job descriptions independently by a hierarchical CNN, and the matching degree is also determined by cosine similarity.
5. **IPJF** (Le et al., 2019): It utilizes multiple labels to represent the propensity of candidates and jobs forming a match.
6. **PJFFF** (Jiang et al., 2020): It learns implicit intentions of the candidates or recruiters from historical accepted and rejected applications.
7. **SHPJF** (Hou et al., 2022): It utilizes both text content from jobs/resumes and search histories from users.
8. **CONFIT** (Yu et al., 2024): It increases the number of training samples using LLM-based data augmentation techniques, and exploits contrastive learning to train a transformer-based encoder.

D Extended Related Work

The Person-Job Fit (PJF) task has received extensive scholarly attention, and it originated from the

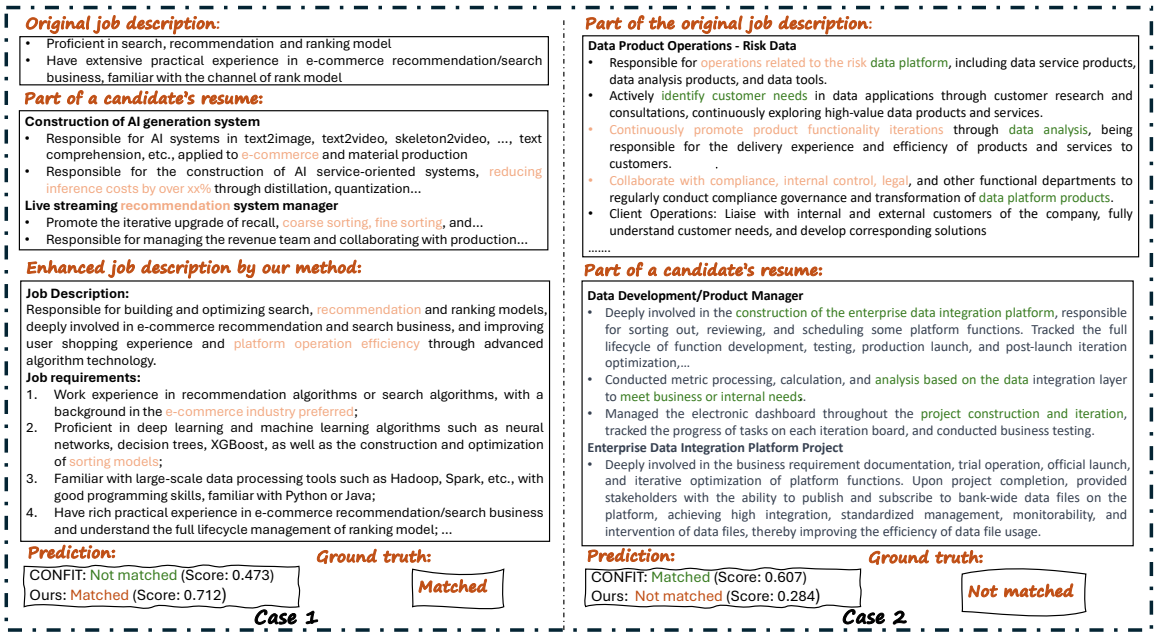


Figure 10: Case study on LLM-based job description augmentation and similar candidate-job pair.

work of Malinowski et al. (2006), which uses the expectation maximization algorithm based on candidate and job opportunity profiles. Traditional studies have approached this task using collaborative filtering (Diaby et al., 2013; Lu et al., 2013; Zhang et al., 2014), while neural networks have been extensively used for this task in recent years (Qin et al., 2020; Wang et al., 2022).

Modeling interactions Some existing methods explore various interaction strategies between candidates and jobs to improve the performance of the PJF model (Le et al., 2019; Fu et al., 2021; Yang et al., 2022; Hou et al., 2022; Yao et al., 2022; Su et al., 2022). For instance, Zheng et al. (2023) formulate reciprocal recommendation as a distinctive sequence matching task and propose to leverage bilateral behavior sequences for dynamic interest modeling on both sides. Zheng et al. (2024) propose a MultI-view Reciprocal Recommender system (MIRROR) which models the users from three different views to capture the user representation corresponding to each view. Compared with the existing methods, the interactions between candidates and jobs of our method are more fine-grained and comprehensive.

E More Experiments

E.1 Impact of Hyperparameters

The ablation study in Table 3 highlights the importance of the category-aware MoE module. Here, we investigate the impact of the number of experts in the MoE module. Results in Figure 9 (a) indicate that five experts yield optimal performance, while increasing the number to seven or nine experts results in performance decline, possibly due to the over-fitting issue. We also carry out experiments to examine how the length of historical interaction sequences affects performance. The findings, illustrated in Figure 9 (b), indicate that performance improves as the sequence length increases up to 20. Beyond this point, longer sequences do not enhance performance and reduce training efficiency. Thus, we set the interaction sequence length to 20 in our experiments.

E.2 More Detailed Ablation Study

Here, we conduct a more detailed ablation study of our category-aware MoE module, as shown in Table 6. *w/o MoE* refers to the model that disables the whole MoE module. *MoE-category* and *MoE-experts* refer to the model removing the category embedding and the experts in the MoE module, while other components in this module remain the same. *Ours (simple match)* is a simple alternative to the MoE module that uses the binary match score (0/1) between the job category and the candidate category to concatenate with other representations. The results of removing the category embedding and experts alone suggest that they both contribute to performance improvements. Additionally, the results of using a simple match to replace the MoE

Table 6: Ablation study of the category-aware MoE module in our method.

Method	AUC	GAUC
Ours (full model)	**0.724**	**0.709**
w/o MoE	0.679	0.665
w/o MoE-category	0.698	0.681
w/o MoE-experts	0.684	0.667
Ours (simple match)	0.694	0.680

module demonstrate that our MoE module is essential for the PJF task.

E.3 Case Study

Here, we present two illustrative cases to demonstrate how our method outperforms the SOTA baseline CONFIT. As shown in Figure 10, *Case 1* shows a low-quality original job description (JD) with limited content and information. CONFIT fails to predict this case accurately (the threshold for transforming the matching score to a binary label is 0.5). In comparison, our method enhances the JD by refining the content and adding some common requirements based on the provided historical resumes and the strong LLM knowledge, leading to accurate predictions. In *Case 2*, the JD and the candidate's resume exhibit several similarities. However, the resume emphasizes data development and product management in enterprise data integration, lacking the operational focus on risk data and customer interaction required by the JD. Additionally, it does not show collaboration with compliance and governance teams, which are key aspects of the job. With the assistance of the category-aware MoE module, our method accurately predicts this case, while CONFIT predicts the wrong label.