In this study, an end-to-end person-to-job post data matching model is constructed, and the experiments for matching people with the actual recruitment data are conducted. First, the representation of the constructed knowledge in the low-dimensional space is described. Then, it is explained in the Bidirectional Encoder Representations from Transformers (BERT) pretraining language model, which is introduced as the encoding model for textual information. The structure of the person-post matching model is explained in terms of the attention mechanism and its computational layers. Finally, the experiments based on the person-post matching model are compared with a variety of person-post matching methods in the actual recruitment dataset, and the experimental results are analyzed.

1. Introduction

In the early studies of person-job fit, researchers generally focused on the relationships between the person and organization (P-O) and person and environment (P-E) [1, 2]. On this basis, some person-job fit models such as the recruitment model based on the P-O fit [3] and the weighted P-O relationship study method [4] were developed. In addition, some researchers proposed focusing on personality and job characteristics, which have an important influence on predicting a person-job fit [5]. With the development of machine learning and deep learning, an increasing number of researchers in the field of machine learning have focused on person-job fit. Some researchers use backpropagation neural networks to design end-to-end fit models, [6] whereas others input manual features and use support vector machines to measure the degree of fit [7]. These methods mostly rely on manual feature construction and require the participation of many experts in related fields. Additionally, these methods cannot fully explore potential information connections. Some advanced methods use the word2vec embedding layer to represent texts and a recurrent neural network or convolutional neural network (CNN) to serve as the feature extractor [8, 9]. However, these advanced methods still need to truncate a large number of sentences and control the overall length of texts to avoid a large input of feature dimensions. Moreover, some researchers have used the idea of a recommendation system to fully investigate the matching between user preferences and items [10, 11] before making recommendations. Other studies have integrated optimal historical data to measure and calculate features [12]. Thus, the method of embedding a priori past resume features has been updated to investigate not only the match between the current resume and the requirements but also other information. Among the studies cited, the bidirectional recommendation system [13, 14] is an important branch in the field of personalized recommendation systems. The bidirectional talent recommendation system matches the preferences and requirements between job applicants and recruiters. This win-win strategy improves the matching accuracy. A similarity technique that combines the explicit and implicit preference data from both job applicants and recruiters has also been proposed [15]. This technique exploits the correlation between the preferences of both job applicants and recruiters and thus has a positive effect on the recommendation results.
Some researchers have also suggested incorporating a large amount of a priori information from the expert team to provide more accurate directional recommendations for some professionals [16, 17]. However, relying on a large amount of expert knowledge leads to a very high usage cost and hinders an automated construction methodology. Related tasks also include predicting the direction of talent flow [18] and investigating the network graphs of the applicant’s talent skills [19]. In these tasks, we explore the importance of the applicant’s talent skills.

Based on these previous studies, we aim to reduce the drawbacks of the word-level embedding model and fully express the correlation. Additionally, we aim to fully express the correlation between job applicants and recruiters and the subject information of texts and incorporate the idea of a recommendation system [20–24]. Therefore, we propose an end-to-end person-job fit model that uses sentence vector representations and integrates knowledge graph information, referred to as KG-DPJF. We follow the methods of Jiang et al. [25]. The model uses a multilayered attention mechanism to fully examine the information interaction between the candidate resume features and the recruitment requirement features. KG-DPJF embeds subject-term information and knowledge graph information to fully explore the relationship between candidates’ text descriptors. In addition, the translating embedding (TransE) method is used to express the resume entity information as a low-dimensional vector and then embed it into the model.

2. Knowledge Representation

Because this study is based on the knowledge-driven person-post matching model, the traditional knowledge representation methods and single-layer neural network model cannot appropriately describe the semantic relationship between entities. The use of the Trans method based on distance measurement can better describe the structural information between entities in space and bring a more accurate embedding of prior knowledge into the model. Then, we define the knowledge map, \( G \), and all triple relationships contained in it.

\[
G = \sum (h, r, t).
\]

When \( G \) is constructed, the entities and relationships are mapped to a low-dimensional continuous vector space using TransE. Suppose \( h, r, \) and \( t \) are the vectors corresponding to the head, relation, and tail, respectively. If there is a triple relationship, it is assumed to indicate that the head entity points to the tail entity through the relationship. Then, the TransE scoring function is

\[
f_j(h, t) = \|h + r - t\|_2^2.
\]

The smaller the scoring function is, the more reliable the triple relationship of \( (h, r, t) \) in the knowledge graph is. In addition, the TransE model has few parameters and low computational complexity, and it still has good performance and scalability in a large-scale sparse knowledge base, which is suitable for fast reasoning task scenarios.

3. BERT Coding

To make the person-post matching model learn the text information contained in the resume and recruitment requirements, it is necessary to use the word embedding matrix to spatially transform the text corpus of the resume and recruitment requirements into a low-dimensional dense spatial vector. If the one-hot representation method is used to represent each word in the sentence vectorially, for example, in the word “deep learning,” assuming that the dictionary composed of all words has only four words, the representation form of one hot is \([1, 0, 0, 0]\). When the number of dictionaries is very large, each word becomes a high-dimensional and sparse tensor; therefore, it is obviously unsuitable to be input to the person-post matching model. In addition, each word can be embedded into a limited low-dimensional space through word2vec. For example, deep learning can be mapped by embedding at the word level into a vector with a dimension of \([200, 4]\). Even if the dictionary contains far more than 1000 words, each word can be mapped to a low dimension by the word2vec embedding matrix as the input to the downstream model. However, statistics show that the number of words of long text items in each original resume is very large, which puts pressure on the model operation, and more text truncation is required during model input, resulting in a loss of matching accuracy of the person-post matching task. Based on the above considerations, the pretraining encoder of BERT is used to encode the sentences of the resume and recruitment requirements, and the powerful information learning capability of the BERT language model is used to encode the long text of the resume and the long text sentences of recruitment requirements with the BERT encoder pretrained based on large Chinese data. The model can further reduce the feature dimensions that need to be input for downstream tasks and retain text information as much as possible. Based on the representation ability of the large-scale pretraining model, all representation information in downstream tasks is fine-tuned to achieve fast and better results.

The number of sentences required by the actual resume and recruitment is not necessarily the same because the content recorded in the text itself is inconsistent, and both can reduce the input feature dimension between the matching models by only outputting the coding vector of the sentence. The coding based on the pretraining model combined with the knowledge extracted from the original corpus can provide more semantic information of the model. In this way, not only is the amount of information maintained but also the model reasoning can be accelerated to a certain extent. The sentence encoding process by BERT can be expressed as

\[
C = \text{bert}(\{s_1, s_2, s_3, \ldots, s_t\}),
\]
where bert represents the encoder based on BERT. Its pretraining model uses the 768-dimensional Chinese pretraining model, where $s_1, s_2, s_3, \ldots, s_t$ is the sentence after text slicing and $t$ represents the total number of sentences. $C \in \mathbb{R}^{t \times d_e}$, where $d_e = 768$ is the fixed coding length. Through the above calculation, the sentence expression embedding matrix $C$ can be produced, which can be input into the downstream task model to replace the word-level embedding matrix input.

4. The Person-Job Matching Model

In the actual recruitment process, the key information recorded in the resume and recruitment requirements often determines the key intention of employment in the recruitment process. The existing person-post matching model directly splices the resume features and recruitment requirements’ features and inputs them into the prediction layer, resulting in low interpretability and a lack of information flow relationship between recruitment and requirements. To solve these problems, we need to fully simulate the recruitment process and build the matching model based on the attention mechanism. The model is divided into feature representation, attention interaction layer, and prediction layer, which is shown in Figure 1.

The feature representation layer must enter the graph embedding matrix, which is the information representation matrix of the graph obtained after the constructed knowledge map of supply and demand is trained by the Trans algorithm. The feature representation layer also includes the feature representation of text information, which represents the sentence-level text information by using the pretraining model BERT. The attention interaction layer uses the attention mechanism to output the set of weighting values between candidate resumes, historical best resumes, and recruitment requirements. The prediction layer accepts the set of weighting values and the feature input of the weighted recruitment requirements and outputs the prediction results through the classifier.

The design of the model is based on the construction process of a deep learning model, including preprocessing, feature representation, multilayer perceptron, and classifier. This includes the preprocessing of the model: as known from the above sections, a BERT encoder can be used to encode the segmented sentences in the input text of the resume and recruitment requirements and output sentence-level vectors. These vectors directly form an embedded sentence-level weight matrix after splicing. In addition, we use Trans to learn the structural information for all the knowledge in the map and output a low-dimensional dense entity embedding weight matrix. In the subsequent matching reasoning process, all the entity vectors of the input text segmentation sentence are extracted, and the entity embedding matrix is formed after splicing.

(1) In the feature representation layer, the input features of the model include three modules. The first is the current input candidate resume, including the work experience segmentation sentence, the project experience segmentation sentence, and the entity. The second is the historical best resume $p_n$, $n$ representing a total of historical resumes. It also includes the work experience segmentation sentence, the project experience segmentation sentence, and the entity. Finally, the resume of recruitment requirements includes job requirements, segmentation sentences, and entities. In addition, other inputs such as entity context and matching score are included. According to the deep learning model, these features are discrete and can be transformed into a high-dimensional independent heat vector. In the construction of the deep learning model, a unique heat vector is transformed into low-dimensional dense features through the embedding layer. The feature item transformation here is sentence-level representation, so each actual resume segmentation sentence is represented by $v$. For example, the feature representation of the current resume is the feature of the resume to be selected $j$, which can be expressed as

$$j = [v_1, v_2, v_3, \ldots, v_d] \in \mathbb{R}^{d \times d_f},$$

where $v$ is the sentence vector after each segmented sentence is encoded. $d_j$ indicates the total number of segmented sentences, and $d_v$ represents the vector dimension after BERT coding. The number of segmented sentences is generally limited; that is, the vector sentences of BERT coding sentences are embedded matrices, which are loaded into the model when the model diagram is running. The representation of historical recruitment characteristics $p$ and the representation of the characteristics of recruitment requirements $q$ can also be expressed as

$$p = [l_1, l_2, l_3, \ldots, l_d] \in \mathbb{R}^{d \times d_i},$$
$$q = [k_1, k_2, k_3, \ldots, k_d] \in \mathbb{R}^{d \times d_i},$$

where $l$ and $k$ are the sentence vectors after the segmented sentences of historical recruitment and recruitment requirements are expressed by coding. $d_p$ and $d_q$ indicate the total number of segmented sentences, and $d_l, d_k$ and $d_q$ are the same representing the vector dimension after BERT coding.

(2) Feature interaction layer: its basic structure is an interactive computing layer based on feature extraction and attention mechanism to obtain the correlation between resume and recruitment requirements by the implicit method. The feature sentence vector matrix after the special representation layer is spliced with the entity vector, in which the entity vector contains the entity and entity context information. The feature of the spliced matrix is extracted using convolutional neural networks (K-convolutional neural networks (KCNNs)), and the output of the CV representation vector with
A fixed dimension is \( c \), and \( n \) is the number of historical resumes \( p \) and current recruitment requirements \( q \). Similarly, after the embedded layer representation, it is input to the convolutional neural network for information extraction, and the historical resume representation is generated, \( u \). The recruitment requirements are \( o \). Among them, KCNN is a convolutional neural network which uses different convolution check features for multiple extraction and superposition. This is expressed as

\[
i = \left[ e_1^h, e_2^h, \ldots, e_m^h \right],
\]

where \( m \) is the number of subconvolutions and convolution kernel size is \( h_m \). This is a preliminary feature extraction method. The parameter data between convolution layer structures are shared, which can not only reduce the memory space but also make the parameter matrix more flexible, which is convenient for attention interactive learning in the later branches. The modeling relationship between the current resume and historical resume is represented by a weighted feature interactive attention mechanism method, and the results express the representation of current resume \( C \). And \( u \) is the correlation weight between \( \tilde{a}_j \); this is regarded as an information enhancement between candidate resumes and historical best resumes. The two historical resume features are equally weighted and averaged here. At the same time, the same attention algorithm is used to express the correlation weight between recruitment requirements and historical resume \( \tilde{a}_q \). \( \tilde{a}_j \) and \( \tilde{a}_q \) can be expressed by the following formula:

\[
\begin{align*}
e_j &= z^T \tanh(W_c c + W_p u + b), \\
\tilde{a}_j &= \frac{\exp(e_j)}{\sum_{i=1}^T \exp(e_j)}, \\
e_q' &= z'^T \tanh(W_q o + W_p' o + b'), \\
\tilde{a}_q' &= \frac{\exp(e_q')}{\sum_{i=1}^T \exp(e_q')},
\end{align*}
\]

where \( W_c \in \mathbb{R}^{d_1 \times d_2} \), \( W_p \in \mathbb{R}^{d_1 \times d_2} \), \( b \in \mathbb{R}^{d_1} \), \( z \in \mathbb{R}^{d_1} \), \( W_q \in \mathbb{R}^{d_1 \times d_2} \), \( W_p' \in \mathbb{R}^{d_1 \times d_2} \), \( b' \in \mathbb{R}^{d_1} \), \( z' \in \mathbb{R}^{d_1} \) are the training parameter matrices. Then, the output weighted attention representation vectors of the current resume are \( c' \). The weighted recruitment representation vector is \( o' \), which can be expressed as the following formula:

\[
\begin{align*}
c' &= \sum_{t=1}^T \tilde{a}_j c, \\
o' &= \sum_{t=1}^T \tilde{a}_q o.
\end{align*}
\]

Here, another layer of attention layer pair is used, \( c' \) and \( o' \), to perform correlation calculation. The output is \( \tilde{a}_c \). The attention layer is mainly used to simulate the mutual attention matching process of the actual recruitment process, fully measure the implicit correlation between the three, and
provide consideration input for the classification layer. $\tilde{a}_c$ can be expressed by the following formula:

$$\tilde{a}_c = \frac{\exp(\tilde{c})}{\sum_{i=1}^{T} \exp(\tilde{c}_i)}$$

(9)

where $W'_i \in \mathbb{R}^{d_{l},d_{l}}$, $W_i \in \mathbb{R}^{d_{l},d_{l}}$, $\tilde{b} \in \mathbb{R}^{d_{l}}$, and $\tilde{z} \in \mathbb{R}^{d_{l}}$. The final resume features after weighted calculation are $\tilde{a}_c$. The calculation process can be expressed as

$$\tilde{a}_c = \sum_{i}^{T} \tilde{a}_i c'.$$

Finally, after calculation, all correlation weights are spliced into $\tilde{u} = [\tilde{a}_c, \tilde{a}_1, \tilde{a}_2]$. $\tilde{u}$ represents $\tilde{a}_c$ and $\tilde{d}$ after splicing; the input that constitutes the prediction layer $\tilde{X} = [\tilde{a}_c, \tilde{d}, \tilde{u}]$. The prediction layer fully takes into account the enhanced resume representation, recruitment requirements’ representation, and all correlations calculated in the model and outputs the matching prediction value.

In the prediction layer, after all the feature representations have passed through the multilayer attention mechanism layer, their weighted feature vectors and related values are output. These vectors are connected through the full connection layer. The hidden layer uses the multilayer fully connected neural network layer and rectified linear unit function to activate. Finally, the output is activated using the sigmoid function. The sigmoid cross-entropy loss function is used to calculate the loss between the logic output and the label. Finally, the feature vector of the input fully connected layer is the output of the attention layer, and the classification label category is ReLU where $y \in \{0, 1\}$. Therefore, the calculation formula of the loss function is

$$L_{\text{target}} = \frac{1}{N} \sum_{(x,y) \in D} \left( y \log(f(x)) + (1-y)\log(1-f(x)) \right),$$

(11)

where $D$ is the training set, with $N$ total of samples, in which $f(x)$ is the prediction result output by the perceptron.

5. Experiments

The data used for the person-post matching experiment are the same as the data collected when the knowledge map was created. However, because of different experimental and data processing methods, the collected experimental data contain matching labels for matching control in the person-post matching experiment.

5.1. Dataset Preprocessing. The experiment uses e-resume data based on the actual recruitment process. Format and extract the desensitized data and clean the data. Referring to the construction method of the deep learning model, the person-post matching model is built using the TensorFlow platform, and the data are converted into the input format of the model. The dataset is stored in the form of resume data corresponding to each position. At the same time, the recruitment mechanism of the recruitment platform and the enterprise employment mechanism of the company should also be considered before the experiment. Recruitment process of the enterprise: the enterprise must roughly select many candidate resumes and put them into the talent folder of the recruitment platform to store resumes or resumes submitted by job-seekers. Then, some suitable personnel are selected to invite for an interview. Finally, only some candidates will pass the interview and be officially hired. This is also the basis for classification. 0 means that the candidate only stops at the stage when the resume enters the talent folder, 1 means only completing the invitation interview stage, and 2 means they pass the invitation interview and get an offer. Note that there are only 0 and 1 labels in the training data; that is, only the data that passed the interview will be given a score of 1, and the rest of the resumes in other stages will be given a score of 0. This is meant to be close to the actual recruitment process and make the model more predictive when combined with the data.

Also, all triplet information extracted from the corpus is trained based on the Trans algorithm. Mapping dataset files of entities, relationships, and triples are constructed, and the data are transformed into a sequence. Then, the data are divided and marked as follows: 80% of the data is used as the test set, 10% of the data is used as the second test set, and the remaining 10% is used as the verification set. The output embedded model dimension is set to 50. When the training is completed, the growth is about 4 h.

After preparing the data required by the model, it starts the training stage. The whole model includes a training and adjustment stage. The two stages can carry out cyclic iteration. After enriching the knowledge map based on continuous new knowledge, it can perform continuous training to generate knowledge representation and then train the person-post matching model to form a complete model application iteration mode driven by knowledge.

During training and prediction, new knowledge is extracted from the text to be predicted. The new knowledge is mapped to the embedded space through the index relationship of the resume ID which corresponds to the entity. The skill keywords described in each resume contain the main idea of the sentence and provide information to support the matching task.

In addition, the optimal historical matching dataset is constructed based on the model design. There is no difference between the optimal historical matching dataset and the ordinary training or test dataset, but the optimal historical matching dataset stores the preset data of the best candidate for the post. The purpose of the setting is to simulate the practice of the recommendation system, embed the historical optimal matching information into the model, and let the model learn the relationship between the predicted resume and the historical optimal matching results to improve the model’s matching performance.

5.2. Experimental Parameters. To realize sentence-level coding, the pretraining BERT model is adopted. So, the online BERT service based on parallel processing can be
used. The information about the service settings for BERT is described in Table 1. The number of parallel runs is set to 4. The Chinese 768 dimension is used in the pretraining model file. The number of parallel runs depends on the number of GPU (graphics processing unit) cores, and this affects the coding speed. Based on the described idea of BERT downstream tasks, the output layer of the model and the output operation of the middle layer of the model were obtained and used as the parameters of the middle layer or output layer of the model, which allows for downstream task improvement training to make the BERT model have better domain semantic expression ability. Here, we first follow the output of the original pretraining BERT code and output the sentence vector. For the person-job matching model parameters, as shown in Table 2, the length of the sentence embedding vector is 768, and the size of entity embedding is 50. In addition, the number of historical optimal recruitment records entered is 4, which depends on the average number of historical matching resumes that were collected for the position. After explaining the setting information required for the experiment and data analysis, the final experiment is generally divided into the following four steps:

1. Data preprocessing: regular expressions are used to remove repeated and continuous labels, large areas of empty lines, and indentations.
2. Build a person-post matching model: the matching model is constructed under the framework of TensorFlow. Then, the training program is constructed, and the file address to save the training is selected.
3. After the hyperparameters are set according to the experimental conditions, the necessary function library is installed, the training program is started, and the model is optimized by the Adam algorithm. The computer has 4 GPUs, and 12 G memory is prepared for the experiments. The model of the graphics card is RTX2080s, and the CPU memory is 64 G.
4. The parameters are adjusted to achieve the best performance of the model and record the experimental data.

5.3. Baseline and Evaluation. This section refers to many existing man-post matching evaluation methods [26–30]. Therefore, in the experimental stage of verifying model performance, traditional methods including the logistic regression (LR) model and the decision tree (DT) method are used to construct the person-post matching model, as well as the existing mainstream person-job fit neural network-RNN (PJFNN-RNN) model based on the RNN. Owing to the test in the industry data, multiple baseline methods need to be created depending on the input mode. Depending on the input mode of the designed person-post matching model, the experiment also specifically constructs a model using the CNN structure to verify the importance of integrating a multilayer attention mechanism. A comparison is then performed between the model of pretraining word2vec embedding layer and the model of sentence-level embedding after BERT coding. Then, the experiment also compares the performance of the person-post matching model using word2vec and BERT coding, and the experiment also refers to a baseline method of text semantic matching, which is measured by measuring the similarity between sentences. Finally, the performance comparison experiment of removing the knowledge map is also conducted.

The idea of constructing the baseline model based on the CNN as the feature extraction layer was inspired by the end-to-end person-post matching work in the existing literature. The model takes the feature representation vector of the text corpus that accepts the input of the resume and recruitment requirements, then uses a CNN with shared features to extract the features, and inputs them to the classification layer after averaging for activation to output the classification probability. Since the input dataset of the model is consistent with the input dataset of the person-post matching model, the BERT coding method can also be used in this experiment to encode the sentence representation, and the word2vec method can also be used to obtain the averaged sentence vector.

After the resume and the required text are represented by the embedded layer, they are input into the CNN, and the resume representation features are output, respectively, $\tilde{S}_f$ and required characteristics $\tilde{S}_q$. These features are input into classifier $D$ and output to the sigmoid layer for prediction through weighted activation function layer $\tilde{y}$ probability of

$$
D = \text{tanh}(W_d [\tilde{s}_j; \tilde{s}_q; \tilde{y} - \tilde{s}_q] + b_d), \\
\tilde{y} = \text{sigmoid}(W_y D + b_y). 
$$

In addition, it also provides a method for sentence semantic matching between long texts required by job posts compared to long texts such as work experience and project experience. The idea comes from the short sentence semantic matching method. First, the sentence is segmented, and then the previously trained word embedding matrix such as word2vec200 and Tencent embedding are used to calculate the representation vector of the text at the level of word segmentation. All the representation vectors form the sentence semantic matrix $V_s$, and then the dimension of the semantic matrix of the text sentence is reduced to output the sentence vector, and then the distance measurement method is used to match the sentences one by one. The purpose is to calculate the distance between the long text data of electronic resumes and the long text data in recruitment requirements listed in job postings one by one and then calculate the final matching score through the overall similarity. This allows for the measurement of the matching degree between resumes and posts. The distance can be measured by cosine similarity, and the calculation formula is as follows:

$$
\cos(\theta) = \frac{\sum_{i=1}^{n} (x_i \times y_i)}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \times \sqrt{\sum_{i=1}^{n} (y_i)^2}}.
$$

where $x_i$ and $y_i$, respectively, represent two sentence vectors that need to calculate cosine similarity. From a spatial perspective, the two sentences can be expressed as two lines.
embedded in the space, taking the coordinate origin as the starting point and pointing to different directions, so that an included angle is formed between the two-line segments. If the included angle is 0 degrees, it means that the two directions of the vector space are the same and the two lines coincide. It also means that the text semantics represented by the two vectors are exactly the same. If the included angle is a 90-degree right angle, it indicates that the directions are completely different, and the semantic results are different. If the included angle is 180 degrees, it means that the direction of the vector is opposite and the semantics are not similar. Therefore, we can represent the text information of the resume and recruitment requirements according to the fully trained word embedding model and then judge the semantic similarity of the vector representation by the size of the included angle. The smaller the included angle, the more similar the sentence semantics. This is also done by segmenting sentences in the above dataset, and experiments are conducted based on the above ideas.

The experimental method is to calculate the similarity between all the sentences based on the resume and all the sentences of the recruitment post, select the highest similarity value in the recruitment requirement text one by one based on all resume sentences, then normalize according to the number of resume sentences, and finally calculate the matching degree. For example, the similarity between sentence 1 and all recruitment-required sentences is calculated in turn, and the highest similarity value is selected as the best matching semantic sentence of the resume sentence and then calculated in turn. Note that, after the sentences of recruitment requirements are matched, they cannot be matched again. Therefore, this is matching from the perspective of the resume.

When comparing the segmentation sentences between the resume and the recruitment requirements, the input data are the resume text information, and the company data are the text information required for recruitment. After embedding space conversion and averaging is performed, this information will be calculated as described above, until all distances are calculated, and the result consistent with the number of service pairs to be matched is output, i.e., it represents the matching degree.

Finally, the performance of the model is compared with the frontier person-post matching model PJFNN-RNN under different structural data to verify the performance of the model. PJFNN-RNN is an end-to-end model based on the RNN as the feature extractor of the resume and required text. Compared with the data analysis in the previous sections, the experiment is based on industry data collected by real enterprises, and there are fixed requirements for the threshold of recruitment regulations. Therefore, in the analysis of the recruitment process, the admission pass rate is set as label 1, and a failure result is set as label 0. The accuracy, the accuracy recall rate, and the $F_1$ value are used as evaluation indicators for the results. Finally, 80% of the dataset is selected as the training set, the next 10% is used to fine-tune the parameters, and the last 10% is used as the test data as the test set. It is worth noting that Table 3 also reflects the problems of data imbalance and label category imbalance. However, because of the differences between civilian posts and technical posts, the data are divided into civilian posts and technical posts. Simultaneously, some sampling methods are limited because of the actual recruitment results. Therefore, the experiment is also based on the actual production application, through cooperation with enterprises, using some of the artificial knowledge to improve some of the data. For example, the description of the CVs of low-level development posts may be closer to that of medium-level development posts, and it is noted that about 3.72% of candidates choose to apply for multiple job postings at the same time. Based on this idea, a small number of similar job postings are sampled and labeled manually, and the datasets are added to the admission and interview file categories.
5.4. Experimental Results and Analysis. The most important and difficult points of the experiment are the processing of resume and recruitment data and the construction of a model training set. Our experiments are based on the industry corpus data collected during the actual recruitment process. The resume and post data files collected in the enterprise recruitment platform after strict desensitization measures are converted into dataset files after preprocessing the data. By analyzing the practical application requirements and related literature, the task is divided into two categories to perform a detailed experimental human work matching task.

First, as the results in Table 4 show, the built embedded knowledge map and BERT-based person-job fit are knowledge graph-based deep-learning-inspired person-job fitting model, KG-DPJF. Table 4 shows the performance of the person-post matching model based on knowledge-driven and multilayer attention mechanisms in the experiment. In addition to the comparison experiment between the model and some baseline models, including DPJF (deep-learning-inspired person-job fitting model), some ablation experiments were also performed. The designed human post matching model integrating the knowledge map and attention mechanism is better than the traditional input feature classifier, and the F1 value is improved by nearly 13%. At the same time, the correlation method that uses multilayer attention calculation to simulate the recruitment process also improves the performance of the model. Compared with the single use of the CNN feature extractor, the F1 value is improved by nearly 6%. In addition, compared with the direct use of pretrained word2vec200 coding, the use of BERT coding can increase the performance. This shows that the BERT model based on unsupervised training of a large number of Chinese corpuses can give the person-post matching model more semantic knowledge for learning, and the BERT can also be fine-tuned again based on the domain’s own data. In addition, Figure 2 also shows the analysis of the training time and also shows that using the BERT-based sentence coding method also has significant advantages in training time, while the method completely based on the word-level input feature extraction layer greatly increases the amount of calculations required, resulting in a longer training time, which can further enlighten the research idea of constructing the person-post matching model in the way of two-stage model structure. In addition, based on the above, because the knowledge is input as the entity embedding layer in the person-post matching model, the knowledge input can be removed for experiments. In the experiments, the model embedded in the knowledge map for job search has better performance than the model without the embedded knowledge map for job search, which also shows that the entity knowledge method embedded with a priori person-post matching information can help the model better understand the meaning of potential skills and keywords behind sentences, express the enhanced correlation between entity words, and combine the rich semantic information of BERT. It can not only reduce the amount of calculation and increase the operation speed but also enhance the matching accuracy. This means that, during the recruitment process, the human resources department will be able to judge whether the candidate is able to meet the standard based on the key information of the text content.

However, it should be noted that only unstructured data are used for testing to match the input and output of the existing person-post matching model; that is, information inputs such as years of work and gender are removed, and...
only work experience and project experience are retained as these discrete characteristics have different processing problems.

In Table 5, we can see the person-post matching model PJFNN-RNN constructed based on the RNN, which is a matching model for person-post text information perception based on the RNN. The PJFNN-RNN model is based on the cosine distance loss function. Therefore, we can compare the performance of models based on unstructured data, that is, long text as the input. From the table, we can see that the KG-DPJF model has some improvement compared to the PJFNN-RNN. The KG-DPJF model with more semantic knowledge performs better than the method that simply relies on long-distance feature extraction and multilayer CNN extraction. At the same time, these methods also have performed better than the methods based on cosine distance rearrangement and DT. It could be that they only rely on distance calculation without learning the feature information between sentence vectors and cannot fully utilize semantic information.

Table 6 begins the experiment to integrate unstructured data, i.e., age, gender, and other characteristics. At the same time, input introduces unstructured knowledge about entities. However, when constructing the input, for these discretionary data, we first use one-hot form coding to measure the performance of different models. The experimental results in Table 6 show that using one-hot coding discrete data can improve the performance of these models. In the context of analysis, it is helpful for matching because it enriches the input of some important matching information, such as gender and education. However, one-hot formal coding has its own shortcomings because it obscures some semantic information, such as the progressive relationship and the importance of educational levels.

The experimental results are shown in Table 7. For discrete data such as structured data, we tried to use BERT to uniformly encode them into a public space rich in semantic information and input them into the person-post matching model for training in combination with their entity information and the remaining unstructured data. The experimental results show that the performance of these models has been improved to a certain extent, and these models can be uniformly encoded into the shared fixed space using BERT in both unstructured data and structured data to provide some information to the language model before training. In particular, the attention mechanism matching model developed in this section based on the knowledge map and BERT can unify the coding input and coding space and learn richer information from the prior knowledge of the knowledge base and the language model. This allows us to enhance the model performance, which also enlightens the future as the performance of BERT domain representation can be adjusted in the first stage. In this manner, the performance of the two-stage person-post matching task is improved.

In this paper, a detailed person-post matching application experiment is conducted. From the above experimental results and analysis, it is found that the person-post matching model based on the attention mechanism designed in this section can achieve an excellent matching effect and realize the matching degree between resumes and recruitment requirements. From the summary of the literature and the analysis of experimental results, it is found that there are still some common defects in the existing experimental tasks of person-post matching:

(1) Inevitable text truncation: in the e-resume data of today’s actual recruitment platform, there are often many sentences in the unstructured data. The number of words displayed in the corpus statistical information map in the experiment is not consistent with the experimental data of other recent literature works. This is because other works usually use the method of truncation or fine extraction before the experiment, restricting the total number of words to a certain range, which is even less than half of the total number of words. In this way, we adapt to the method of not using a pretraining vector representation and feature extraction and hope to fully learn the semantic information in sentences. This study discusses how to make full use of pretraining sentence information, pretraining model, and prior knowledge to solve this problem.

(2) Matching defects caused by data imbalance: in real electronic resume data, the data often consist of semistructured data, while other works usually transform items such as gender and education into discrete data. This is because their values have the ability to possibly change the perspective of the whole recruitment such as education: masters and

| Model            | Accuracy | Precision | Recall  | F1    |
|------------------|----------|-----------|---------|------|
| Logistic regression | 0.5091   | 0.4493    | 0.4473  | 0.4627|
| Decision tree    | 0.5369   | 0.4478    | 0.4615  | 0.4646|
| Cosine distance  | 0.5275   | 0.5281    | 0.5034  | 0.5169|
| CNN and word2vec | 0.5632   | 0.5001    | 0.5385  | 0.5185|
| CNN and BERT     | 0.5736   | 0.5147    | 0.5465  | 0.5383|
| PJFNN-RNN        | 0.6023   | 0.5716    | 0.5864  | 0.5789|
| KG-DPJF          | 0.6169   | 0.5810    | 0.6047  | 0.5926|

| Model             | Accuracy | Precision | Recall  | F1    |
|-------------------|----------|-----------|---------|------|
| Logistic regression | 0.5955   | 0.4973    | 0.5034  | 0.5040|
| Decision tree     | 0.6247   | 0.5159    | 0.4800  | 0.4973|
| PJFNN-RNN         | 0.6333   | 0.6511    | 0.5689  | 0.6079|
| KG-DPJF           | 0.6562   | 0.6467    | 0.6018  | 0.6286|

| Model            | Accuracy | Precision | Recall  | F1    |
|------------------|----------|-----------|---------|------|
| Logistic regression | 0.6023   | 0.4956    | 0.5177  | 0.5063|
| Decision tree    | 0.6274   | 0.5138    | 0.4967  | 0.5050|
| PJFNN-RNN        | 0.6528   | 0.6274    | 0.5953  | 0.6109|
| KG-DPJF          | 0.6522   | 0.6483    | 0.6125  | 0.6298|

Table 5: The comparative experiment—no structured data.

Table 6: The comparative experiment with structured data one hot.

Table 7: The comparative experiment combined with structured data BERT.
college degree. These ordered datasets cannot be well solved by deep learning methods in the actual recruitment data. In addition, there is also a relationship between recruitment discrimination and data samples. This usually reflects inequality in recruitment. For example, statistically, there are more men than women programmers. The model should understand and avoid these datasets. However, in the experiment in Table 7, instead of massaging the data, the data are regarded as short sentences, encoding their semantic information, embedding entity information, and transferring the expressed theme learning to the matching model. In the future, we will investigate whether the knowledge map provides effective information and internal relationship for the processing of these ordered data.

6. Conclusion

Based on the knowledge map of supply and demand, this study embeds and represents knowledge into a low-dimensional dense space. The research shows that the person-post matching model based on the knowledge-driven and linguistic model has better performance and more interpretability than the existing person-post matching methods, which can provide some clues for the future work on person-post matching.

Data Availability

All the data generated or analyzed during this study are included within this article.

Disclosure

Xiaowei Wang and Zhenhong Jiang are the co-first authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Open Research Fund Program of Data Recovery Key Laboratory of Sichuan Province (Grant no. DRN19012).

References

[1] S. A. Carless, "Person-job fit versus person-organization fit as predictors of organizational attraction and job acceptance intentions: a longitudinal study," Journal of Occupational and Organizational Psychology, vol. 78, no. 3, pp. 411–429, 2005.
[2] J. D. Werbel and S. W. Gilliland, "Person–environment fit in the selection process," Research in human resources management, vol. 17, pp. 209–243, 1999.
[3] C. Zhang and Y. Ye, "A recruitment mode based on person-organization match," Research on Financial and Economic Issues, vol. 3, pp. 94–98, 2006.
[4] T. Sekiguchi and V. L. Huber, "The use of person-organization fit and person-job fit information in making selection decisions," Organizational Behavior and Human Decision Processes, vol. 116, no. 2, pp. 203–216, 2011.
[5] K. H. Ehrhart, "Job characteristic beliefs and personality as antecedents of subjective person-job fit," Journal of Business and Psychology, vol. 21, no. 2, pp. 193–226, 2006.
[6] Z. Yuan and S. Lu, "Application in person-job fit evaluation of BP neural network in knowledge workers," Journal of Wuhan University of Technology, vol. 32, no. 3, pp. 515–518, 2010.
[7] Y. Bing and J. Fei, "Application in person-job fit evaluation of support vector machine," Journal of Central South University of Forestry & Technology (Social Sciences), vol. 5, no. 6, pp. 92–94, 2011.
[8] C. Qin, H. Zhu, T. Xu, C. Zhu, and L. Jiang, "Enhancing person-job fit for talent recruitment: an ability-aware neural network approach," in Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 25–34, Ann Arbor, MI, USA, 2018.
[9] C. Zhu, H. Zhu, H. Xiong et al., "Person-job fit: adapting the right talent for the right job with joint representation learning," ACM Transactions on Management Information Systems, vol. 9, no. 3, pp. 12.1–12.17, 2018.
[10] Y. Lu, S. Ingram, and D. Gillet, "A recommender system for job seeking and recruiting website," in Proceedings of the 22nd International Conference on World Wide Web, pp. 963–966, New York, NY, USA, 2013.
[11] A. Gionis, D. Gunopulos, and N. Koudas, "Machine learned job recommendation," in Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 325–328, Chicago, IL, USA, 2011.
[12] D. Wu, L. Zhou, and H. Lin, "Design and implementation of job recommendation system for graduates based on random walk," Journal of Guangxi Normal University, vol. 29, no. 1, pp. 179–185, 2011.
[13] T. Weitzel, "Matching people and jobs: a bilateral recommendation approach," in Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS’06), 2006.
[14] T. Chung, J. Kay, I. Koprisnka, L. Pizzato, T. Rej, and K. Yacefculum, "Reciprocal recommenders," in Proceedings of the 8th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (UMAP’2010), pp. 1–12, Big Island, HI, USA, 2010.
[15] H. Yu, C. Liu, and F. Zhang, "Reciprocal recommendation algorithm for the filed of recruitment," Journal of Information and Computational Science, vol. 8, no. 16, pp. 4061–4068, 2011.
[16] S. Trewin, "Knowledge-based recommender systems," Encyclopedia of Library and Information Science, vol. 69, pp. 69–180, 2000.
[17] J. D. Velásquez and V. Palade, "Building a knowledge base for implementing a webbased computerized recommendation system," The International Journal on Artificial Intelligence Tools, vol. 16, no. 5, pp. 793–828, 2007.
[18] L. Zhang, H. Zhu, T. Xu et al., "Large-scale talent flow forecast with dynamic latent factor model?" in Proceedings of the 2019 World Wide Web Conference, pp. 2312–2322, San Francisco, CA, USA, 2019.
[19] T. Xu, H. Zhu, C. Zhu, and H. Xiong, "Measuring the popularity of job skills in recruitment market: a multi-criteria approach," 2018, https://arxiv.org/abs/1712.03087.
[20] T. Zhang, B. Liu, D. Niu, K. Lai, and Y. Xu, "Multiresolution graph attention networks for relevance matching," in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 933–942, Turin, Italy, 2018.
[21] Z. Lyu, Y. Dong, C. Huo, and W. Ren, “Deep match to rank model for personalized click-through rate prediction,” in Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 2020.

[22] L. Peng, H. Liu, Y. Nie, Y. Xie, X. Tang, and P. Luo, “The transnational happiness study with big data technology,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 1, pp. 1–12, 2021.

[23] Z. Zhao, C. K. Ahn, and H.-X. Li, “Dead zone compensation and adaptive vibration control of uncertain spatial flexible riser systems,” IEEE, vol. 25, no. 3, pp. 1398–1408, 2020.

[24] Z. Zhao and Z. Liu, “Finite-time convergence disturbance rejection control for a flexible timoshenko manipulator,” IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 1, pp. 157–168, 2021.

[25] Z. Jiang, L. Peng, and L. Shi, “Person-Job Fit model based on sentence-level representation and theme-word graph,” in Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 1902–1909, Compiègne, France, 2021.

[26] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, https://arxiv.org/abs/1508.01991.

[27] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” Advances in Neural Information Processing Systems, vol. 2, pp. 2787–2795, 2013.

[28] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” 2018, https://arxiv.org/abs/1810.04805.

[29] C. Qin, H. Zhu, T. Xu et al., “An enhanced neural network approach to person-job fit in talent recruitment,” ACM Transactions on Information Systems, vol. 38, no. 2, pp. 1–33, 2020.

[30] C. Zhu, H. Zhu, H. Xiong, and P. Ding, “Recruitment market trend analysis with sequential latent variable models,” in Proceedings of the 22nd ACM SIGKDD International Conference, pp. 383–392, San Francisco, CA, USA, 2017.