Person-Job Fit: Adapting the Right Talent for the Right Job with Joint Representation Learning

CHEN ZHU and HENGSHU ZHU, Talent Intelligence Center, Baidu, Inc., China
HUI XIONG, Business Intelligence Lab, Baidu Research, Talent Intelligence Center, Baidu, Inc., China
CHAO MA, FANG XIE, PENGLIANG DING, and PAN LI, Talent Intelligence Center, Baidu, Inc., China

12

Person-Job Fit is the process of matching the right talent for the right job by identifying talent competencies that are required for the job. While many qualitative efforts have been made in related fields, it still lacks quantitative ways of measuring talent competencies as well as the job's talent requirements. To this end, in this article, we propose a novel end-to-end data-driven model based on a Convolutional Neural Network (CNN), namely, the Person-Job Fit Neural Network (PJFNN), for matching a talent qualification to the requirements of a job. To be specific, PJFNN is a bipartite neural network that can effectively learn the joint representation of Person-Job fitness from historical job applications. In particular, due to the design of a hierarchical representation structure, PJFNN can not only estimate whether a candidate fits a job but also identify which specific requirement items in the job posting are satisfied by the candidate by measuring the distances between corresponding latent representations. Finally, the extensive experiments on a large-scale real-world dataset clearly validate the performance of PJFNN in terms of Person-Job Fit prediction. Also, we provide effective data visualization to show some job and talent benchmark insights obtained by PJFNN.

CCS Concepts: • **Information systems → Wrappers (data mining)**; *Content analysis and feature selection*;

Additional Key Words and Phrases: Recruitment analysis, joint representation learning

ACM Reference format:
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 Trans. Manage. Inf. Syst.* 9, 3, Article 12 (September 2018), 17 pages.
https://doi.org/10.1145/3234465

1 INTRODUCTION

Person-Job Fit refers to the process of matching the right talent for the right job through effectively linking talent competencies to job requirements. Many studies have shown that Person-Job Fit can be related to productivity and commitment (Robbins 2001). However, since there are huge numbers of job candidates and job postings available on the Internet, the gap between talent and

Authors' addresses: C. Zhu and H. Zhu (corresponding author), Talent Intelligence Center, Baidu, Inc., Beijing, China; emails: {zhuchen02, zhuhengshu}@baidu.com; H. Xiong (corresponding author), Business Intelligence Lab, Baidu Research, Beijing, China; email: hxiong@rutgers.edu; C. Ma, F. Xie, P. Ding, and P. Li, Talent Intelligence Center, Baidu, Inc., Beijing, China; emails: {machao13, xiefang, dingpengliang}@baidu.com, lp970625@mail.ustc.edu.cn.

2158-656X/2018/09-ART12 $15.00
https://doi.org/10.1145/3234465

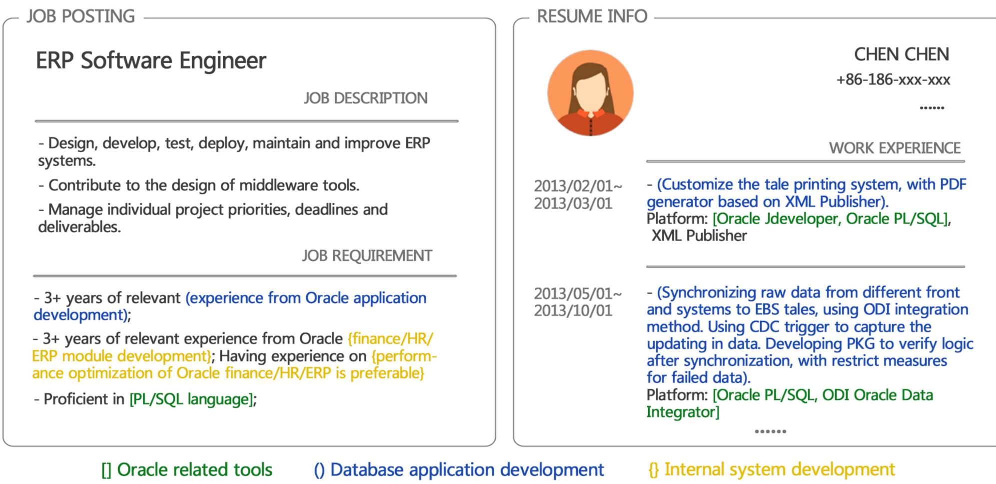


Fig. 1. A motivating example of Person-Job Fit.

job opportunities has been increasing regardless of the importance of Person-Job Fit. For example, as of 2015, there were more than 400 million people available at LinkedIn (linkedIn Wikipedia 2017). Meanwhile, it is reported that recruiters need to spend on average 52 days and $4,000 for filling an open job position with the right talent (Factbook 2016).

In the literature, the research related to Person-Job Fit is usually formulated as the problem of job/candidate recommendations (Malinowski et al. 2006; Paparrizos et al. 2011; Zhang et al. 2016, 2015) and talent sourcing (Xu et al. 2016; Zhu et al. 2016). However, a significant challenge along this line is how to quantitatively measure talent competencies as well as the job's talent requirements. Figure 1 shows a motivating example of this article. In the figure, a job posting and the resume of a successful applicant are listed. Specifically, there are three major requirements for this job that represent different needs of work responsibility, namely, *"Oracle related tools"* (marked with "[ ]"), *"Internal system development"* (marked with "{}"), and *"Database application development"* (marked with "()"). For the applicant, the work experiences clearly validate that she has a strong background in Oracle development and database applications, which match the job requirements well. As a result, while the applicant has no experience in internal system development, the employers believe she fits the job in general. Indeed, while the above inspection of Person-Job fitness is apparent for those experts in this field, most recruiters, who are responsible for sourcing resumes, are not very familiar with the related knowledge and cannot justify the fitness efficiently. If we can provide an explicable way of finding Person-Job Fit, it will help to improve the efficiency of recruiters.

To this end, in this article, we propose a novel end-to-end data-driven model based on a Convolutional Neural Network (CNN), namely, the Person-Job Fit Neural Network (PJFNN), to match the right talent for the right job by learning the joint representation of Person-Job fitness from historical job applications. Specifically, it assumes that both job postings and resumes can be projected onto a shared latent representation, along with each requirement item in job postings and each work experience item in resumes. Meanwhile, each job posting (requirement item) and its corresponding resumes (work experience items) should be similar in the representation space. With the help of PJFNN, we can not only estimate whether a candidate fits a job but also identify which specific requirement items in the job posting are satisfied by the candidate. The major contributions of this article can be summarized as follows:

- We formulate Person-Job Fit as a joint representation learning problem, which aims to match a candidate's work experience to job requirements and thus provides a new research paradigm in talent recruitment.
- We propose a novel CNN-based end-to-end model, namely, PJFNN, for the proposed problems, which projects both job postings and candidate resumes onto a shared latent representation by joint representation learning from historical job applications, along with each requirement item in jobs and each work experience item in resumes.
- The proposed method has been practiced in real-world scenarios. The extensive experiments on a large-scale real-world dataset clearly validate the performance of our approach in terms of Person-Job Fit prediction. Besides, we design effective data visualization to show some job and talent benchmark insights obtained by our method.

Overview. The remainder of this article is organized as follows. Section 2 provides a brief review of related works. Section 3 introduces the details of our model PJFNN. In Section 4, we report the evaluation results based on a real-world dataset. Finally, we conclude the article in Section 5.

2 RELATED WORK

In this section, we will briefly introduce some works related to this article. According to the research problem and the technology used in this article, the related works can be grouped into two categories, namely, Person-Job Fit and text mining with neural network.

2.1 Person-Job Fit

Recruitment is a core function of human resource management. The traditional effort to measure the fitness between employees and job positions is best articulated in Personality-Job Fit theory (Robbins 2001), which identifies six personality types (i.e., Realistic, Investigative, Artistic, Social, Enterprising, and Conventional) (Holland 1973) and proposes that the fitness between personality type and occupational environment determines the job satisfaction and turnover. Although this theory has been widely accepted in academia and the industry, how to precisely measure personality and fitness between jobs and jobs seekers is a vital problem every recruiter will face. Traditionally, a person's personality profile is measured by a well-designed inventory questionnaire, and the fitness is determined by recruiters without objective metrics. Obviously, this method is subjective and often leads to biases.

Due to the explosion of online recruitment markets, recruitment analysis has been attracting more and more attention (Qin et al. 2018; Xu et al. 2018; Zhu et al. 2016) from researchers. Traditional efforts tend to treat Person-Job Fit as a job/candidate recommendation problem. In Malinowski et al. (2006) tried to find a good match between talents and jobs by two distinct recommendation systems. Then Diaby et al. (2013) presented a content-based recommender system for recommending jobs to Facebook and LinkedIn users. Lu et al. (2013) exploited job and user profiles and the actions undertaken by users to propose a hybrid recommender system. To address the challenge that job applicants do not update the user profile in a timely manner, Hong et al. extended users' profile dynamically by job application records and their behaviors for better recommendation (Hong et al. 2013). Zhang et al. (2015) leveraged collaborative filtering and some background information to recommended suitable jobs for candidates.

Recently, some researchers tried to study the Person-Job Fit problem from novel perspectives. For example, Paparrizos et al. (2011) exploited all historical job transitions as well as the data associated with employees and institutions to predict an employee's next job transition. Zhang et al. (2016) created the generalized linear mixed model (GLMix), a more fine-grained model at the user or item level, for the LinkedIn job recommender system and generated 20% to 40% more job applications. Li et al. (2016) proposed an approach to apply standardized entity data to improve job search quality in LinkedIn and to make search results more personalized. In Cheng et al. (2013), job information is extracted from social networks and used to construct an intercompany job-hopping network, which clearly demonstrates the flow of talents. Xu et al. (2018). Lin measured the popularity of job skill by modeling the generation of a skill network. Lin et al. (2017) proposed collaboratively modeling both textual (e.g., reviews) and numerical information to learn the latent structural patterns of companies. Shen et al. (2018) tried to improve recruitment efficiency by intelligent interview assessment.

Although the above studies have explored different research aspects of Person-Job Fit, few of them can provide comprehensible reasons behind their job/candidate recommendation results, which benefit both employers and job seekers.

2.2 Text Mining with Neural Network

Because both resumes and job postings are textual data, Person-Job Fit can be treated as a match between texts. Recently, the deep neural network (DNN) has become one of the hottest techniques in this field due to its good performance.

DNNs applied in text mining can be generally divided into two categories, CNN and recurrent neural network (RNN). CNN aims at modelling hierarchical relationships and extracting local semantics. The effort of applying CNN in text mining can date back to Kalchbrenner et al. (2014), where Kalchbrenner et al. creatively proposed the Dynamic Convolutional Neural Network (DCNN) to model sentences. Then many researchers began to solve NLP problems by CNN. Kim (2014) demonstrated that CNN, even just using a convolutional layer, also performs remarkably well in many NLP tasks. Different from CNN, RNN is good at modeling sequence relationships and finding global semantics. Thus, it performed very well in sequential labeling problems in text mining, such as machine translation (Sutskever et al. 2014) and contextual parsing (Vinyals et al. 2015).

In text mining, both machine translation and multilingual word embedding study aligned data and thus are similar to our problem. Neural machine translation, which aims to build a neural network to read a sentence and output a correct translation, is a newly emerging approach to machine translation. Most of these works belong to a family of Encoder-Decoder. For example, Sutskever et al. (2014) proposed to use LSTMs to map sentences to sentences. Cho et al. (2014) proposed a novel gated recursive convolutional neural network for the Encoder-Decoder-based translation framework. Devlin et al. (2014) proposed a novel formulation for a neural network joint model and yielded strong empirical results. On the other hand, multilingual word embedding aims to map words from different languages into a shared latent space. Lauly et al. (2014) tried to use an autoencoder to learn word representations. Similarly, Hermann and Blunsom (2014) tried to assign similar embeddings to aligned sentences for learning semantic representations without aligned words.

Obviously, both neural machine translation and multilingual word embedding are good solutions for learning the relationships between aligned data, which also exist in the Person-Job Fit problem. However, most of these methods, such as Hermann and Blunsom (2014), need aligned relationships in the sentence level or even word level, which is not available in our problem. In this article, we modify some state-of-the-art ideas in the above works to adapt our problem and propose a novel model to link talents to jobs.

3 MODEL DESCRIPTION

In this section, we will first introduce some research preliminaries and then explain the proposed model, PJFNN, in detail.

3.1 Preliminaries

In practice, the recruitment data usually consist of three parts, job postings, resumes, and job application records. Specifically, a job posting contains *job content* (e.g., job duty) and *job requirement* that consists of several requirement items (e.g., qualifications about skills or experiences). A resume contains a candidate's *profile* (e.g., age and gender) and work experience that consists of several work experience items (e.g., project experiences in previous/current companies). To simplify our problem, we assume that a job can be represented by its job requirements, and the work experiences of a candidate can mainly reflect his or her competency. Thus, we can formulate Person-Job Fit as matching a candidate's work experiences to job requirements. In particular, we regard a job posting as a set of requirement items and a resume as a set of work experience items. Those job application records naturally provide labeled data for us. Please note that there exist overlaps in job application data. In other words, a candidate can apply for several jobs, and of course a job can be applied for by many candidates. Thus, in different job applications, the jobs may be the same. So may those resumes.

We assume that there is a shared latent representation for job postings and resumes. Accordingly, each job posting and each resume can be represented by a vector on the latent representation. Therefore, a candidate fits a job well only if its vectors on the representation are similar. Similarly, requirement items and work experience items can also be projected onto this representation, and the distances between them reflect the corresponding fitness. To facilitate understanding, we still take the case in Figure 1 as an example. Specifically, *"Oracle related tools," "Internal system development,"* and *"Database application development"* are three latent factors that jointly form the representation of the given job posting. We can observe that the two work experiences in the resume endorse that the candidate fits requirements 1 and 3 of this job posting well, but misses requirement 2 related to *"Internal system development."* Therefore, compared with requirement 2, the latent vectors of both the work experience items should be more similar to those of the other requirements. Meanwhile, considering this candidate has met two of three requirements, the latent vectors of the job posting and the resume should be relatively similar.

Formally, in this article, the successful job application records are represented by set A, and each record $a_i \in A$ is a pair (j_i, r_i), where j_i and r_i are the corresponding job posting and resume, respectively. Job posting j_i is a set of items $j_i = \{j_{i,0}, j_{i,1}, \ldots, j_{i,n_{j_i}}\}$, where n_{j_i} is the number of requirement items in j_i. Resume r_i is also a set of work experience items $r_i = \{e_{i,0}, e_{i,1}, \ldots, e_{i,n_{r_i}}\}$. Besides, we use F to represent the set of failed job applications and each record $f_i \in F$ is a pair (j_f, r_f). We define $\mathbf{v^{j_i}}$ as the latent vector of job posting j_i and $\{\mathbf{v^{j_i}_n}\}_n$ as the latent vectors of requirement items in this job posting. Furthermore, the latent vectors $\mathbf{v^{r_i}}$ and $\{\mathbf{v^{r_i}_n}\}_n$ are defined for resumes in a similar way.

3.2 Person-Job Fit Neural Network

To match the right talent for the right job, we propose a CNN-based model, PJFNN, for effectively learning the joint representation of Person-Job fitness. Its architecture and hyperparameters are shown in Figure 2. Generally, PJFNN is a bipartite neural network that can be separated into two parallel parts, namely, *job part* and *resume part*. With this design, PJFNN can project job postings and candidate resumes onto a shared latent representation, respectively. In the following, we will explain the job part in detail and then briefly introduce the resume part by pointing out the differences between them.

Job Part: As mentioned above, each job is regarded as a set of requirement items. Specifically, given an item with $|S|$ words, we take the embedding $\mathbf{w_i} \in \mathbf{R^d}$ for each word and construct the

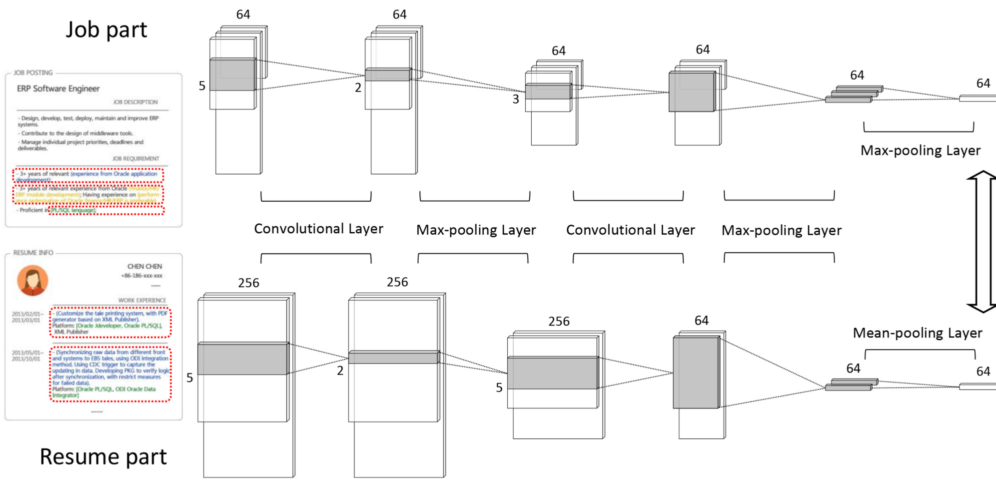


Fig. 2. An illustration of the architecture of PFJNN, which can be separated into two parts (e.g., job part and resume part). Two 1-dimensional convolutional layers are applied on each requirement (work experience) item. Each convolutional layer is followed by a max-pooling layer, where the stride of the first max-pooling layers is 2, and the size of the second max-pooling layer is set as the length of its input to map a requirement (work experience) item into a vector. The corresponding hyperparameters are shown in this figure. Finally, the vectors of all requirement (work experience) items are projected onto a vector by a max-pooling layer (mean-pooling layer) to represent the corresponding job posting (resume).

corresponding matrix $S \in R^{d \times |S|}$:

$$S = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s]. \tag{1}$$

The item matrix is the input layer of PJFNN, and a job posting j_i can be represented as

$$j_i = [S_1, S_2, \ldots, S_{n_{j_i}}]^\top, \tag{2}$$

where n_{j_i} is the number of requirement items in job posting j_i.

Then we apply two 1-dimensional convolutional layers on the input layer. The one-dimensional convolution is an operation between a vector of weights $\mathbf{m}$ and a vector of input viewed as a sequence S. The idea behind this operation is to take the dot product of the weight parameter $\mathbf{m}$ with each m-gram in a sequence to produce output sequence $\mathbf{C}$, i.e.:

$$c_i = \mathbf{m}^\top \mathbf{S}_{\mathbf{i\text{-}|m|+1:i}}. \tag{3}$$

Because one-dimensional convolution can deal with an unfixed-length sequence, it is widely used to model text (Kalchbrenner et al. 2014).

To reduce the training cost, we apply Batch Normalization (Ioffe and Szegedy 2015) on the outputs of one-dimensional convolutional layers. Batch Normalization is a mechanism for dramatically accelerating the training of deep networks, which also eliminates the importance of initialization. Furthermore, the Batch Normalization is followed by a Rectified Linear Unit (ReLU) layer (Nair and Hinton 2010) and a one-dimensional max-pooling layer. Please note that the size of the second max-pooling layer is the length of its input, so that a sentence can be mapped into a vector, $\mathbf{v}_\mathbf{n}^{\mathbf{j_i}}$. With this part of PJFNN, job posting j_i can be transformed into

$$\left[\mathbf{v}_\mathbf{0}^{\mathbf{j_i}}, \mathbf{v}_\mathbf{1}^{\mathbf{j_i}}, \ldots, \mathbf{v}_{\mathbf{n_{j_i}}}^{\mathbf{j_i}}\right]^\top. \tag{4}$$

Then we use a max-pooling layer to integrate them into $\mathbf{v}^{\mathbf{j_i}}$ as

$$\mathbf{v}^{\mathbf{j_i}} = \left[\mathbf{max}\left(\mathbf{v}^{\mathbf{j_i}}_{*,\mathbf{0}}\right), \mathbf{max}\left(\mathbf{v}^{\mathbf{j_i}}_{*,\mathbf{1}}\right), \ldots, \mathbf{max}\left(\mathbf{v}^{\mathbf{j_i}}_{*,\mathbf{l}}\right)\right], \tag{5}$$

where l is the length of latent requirement vectors, and $v^{j_i}_{*,k}$ is the vector of all representations of items in dimension k.

Resume Part: Generally, the resume part of PJFNN is similar to the job part. The difference is how to integrate item representations into resume representations. Here, we propose to use a mean-pooling layer for modeling them and set

$$\mathbf{v}^{\mathbf{r_i}} = \left(\mathbf{v}^{\mathbf{r_i}}_{\mathbf{0}} + \mathbf{v}^{\mathbf{r_i}}_{\mathbf{1}} + \cdots + \mathbf{v}^{\mathbf{r_i}}_{\mathbf{n_{r_i}}}\right)/\mathbf{n}. \tag{6}$$

To get a reasonable shared low-dimension representation, we minimize

$$Loss(A) = \sum_{i=1}^{|A|} D(j_i, r_i), \tag{7}$$

where $D(j_i, r_i)$ is the distance between v^{j_i} and v^{r_i}. However, if we directly minimize the loss function above, PJFNN would learn to reduce all parameters to zero. To address this issue, we simultaneously minimize the distances between representations of resumes v^{r_i} and job postings v^{j_i} in the successful job applications and maximize the distances between representations of resumes v^{r_f} and job postings v^{j_f} in the failed job applications. It is obvious that how to select the failed application set has a large impact on the performance of our model. We will discuss this in detail in Section 4. Besides, we also add L_2 regularization of all parameters in PJFNN θ to avoid overfitting. Therefore, the objective function is formulated as

$$min\left(\sum_{i=1}^{|A|} D(j_i, r_i) - \sum_{f=1}^{|F|} D(j_f, r_f) + \lambda\|\theta\|^2\right). \tag{8}$$

In this article, we select cosine similarity to measure the distance between latent representations,

$$D(j_i, r_i) = -\frac{\mathbf{v}^{\mathbf{j_i}} \cdot \mathbf{v}^{\mathbf{r_i}}}{\|\mathbf{v}^{\mathbf{j_i}}\|\|\mathbf{v}^{\mathbf{r_i}}\|}. \tag{9}$$

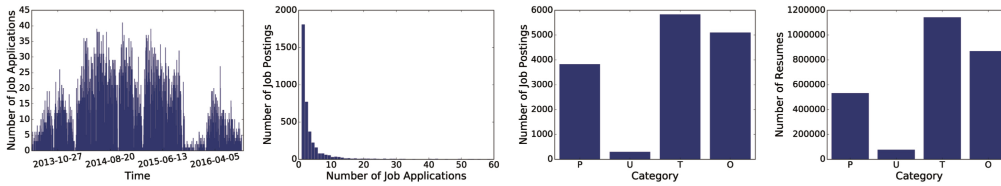
Finally, the objective function can be rewritten as

$$min\left(\sum_{i=1}^{|A|} -\frac{\mathbf{v}^{\mathbf{j_i}} \cdot \mathbf{v}^{\mathbf{r_i}}}{\|\mathbf{v}^{\mathbf{j_i}}\|\|\mathbf{v}^{\mathbf{r_i}}\|} + \sum_{f=1}^{|F|} \frac{\mathbf{v}^{\mathbf{j_f}} \cdot \mathbf{v}^{\mathbf{r_f}}}{\|\mathbf{v}^{\mathbf{j_f}}\|\|\mathbf{v}^{\mathbf{r_f}}\|} + \lambda\|\theta\|^2\right). \tag{10}$$

The objective function is optimized by the Adam algorithm (Kingma and Ba 2014).

Discussion (CNN vs. RNN). In text mining, most DNNs can be classified into two categories, CNN and RNN. There has been a lot of work on comparing their performance on text mining tasks (Vu et al. 2016; Yin et al. 2017). Generally, it is believed that CNN is good at modeling hierarchical relationships and local semantics. Meanwhile, RNN focuses on sequential dependency and global semantics. In PJFNN, we propose to use CNN rather than RNN for modeling textual data, which is because CNN can better capture the hierarchical relationships and local semantics between a job posting (resume) and its requirement (work experience) items. Actually, in the recruitment data, each item only consists of short sentences and limited keywords, and the sequential dependency between different items is not significant. Therefore, we think CNN is a more effective approach for modeling the textual data in Person-Job Fit.

Discussion (Job vs. Resume). In PJFNN, we propose leveraging the max-pooling for job part modeling and the mean-pooling for resume part modeling. Indeed, we think each dimension of the shared latent representation can reflect a specific latent aspect of expertise in some ways.



(a) (b) (c) (d)

Fig. 3. The distribution of (a) the number of successful job applications with respect to time spans, (b) the number of job postings with respect to the number of their successful job applications, (c) the number of job postings with respect to different categories, and (d) the number of resumes with respect to different categories.

Intuitively, since the job posting is usually well formatted, different requirement items in a job posting usually represent different aspects of expertise independently. In contrast, each work experience item of candidates is often a mixture of expertise, since candidates usually want to thoroughly demonstrate their abilities in the work experience. Just as the example shown in Figure 1, three requirement items represent three different needs, containing *"Oracle related tools," "Internal system development,"* and *"Database application development,"* while both of the candidate's work experience items meet the first and third needs of this job.

4 EXPERIMENTS

In this section, we will evaluate the performance of our model on a large-scale real-world recruitment dataset.

4.1 Experimental Setup

The dataset used in the experiments is the historical job application records of a large high-tech company in China, which ranges from 2013 to 2016. It contains more than 2 million resumes and 15,039 job postings, while there are only 31,928 successful job applications. Indeed, the low admit rate (≈1%) clearly validates the importance of Person-Job Fit in talent recruitment. To avoid biases, we removed intern applications and resumes/job postings without any information. After that, the filtered dataset contains 12,007 successful job applications. Specifically, we demonstrate some basic statistics of our dataset in Figure 3. From Figure 3(a), we can observe that the number of recruitments is relatively steady, except from October 2015 to February 2016, where the dramatic decline is due to the change of recruitment policy (i.e., partial hiring freeze). Meanwhile, the number of job postings with respect to the number of their successful job applications roughly follows a long-tail distribution according to Figure 3(b). In our dataset, each job posting/resume is classified into a job category by work content, which can be *Technology (T)*, *Product (P)*, *User interface/experience (U)*, or *Others (O)*. From Figure 3(c) and Figure 3(d), we can find that the recruitment demand of category T is the largest, followed by P and O.

What should be mentioned is that words should be represented by vectors first in our model. Thus, we first used the Skip-gram Model (Mikolov et al. 2013) to encode words of resumes and job postings into 64-dimension and 256-dimension vectors, respectively. Please note that, to get a generalized embedding, we trained the Skip-gram on the entire dataset rather than the filtered one. And the word embedding would not be changed during training of our model.

Besides, all of the applicants in successful applications must go through a rigorous interview, and thus they can be directly used as positive samples. But as for these failed job applications, the reasons for these failures are various. For example, some failed applications may be just due to the low pay/benefits rather than the unfitness of their work experiences and job requirements. These applications, which were labeled as failure, should be regarded as successful applications. Thus, to guarantee the effectiveness of representation learning, we generated negative samples by replacing job postings in successful applications with randomly selected job postings, instead of extracting negative samples from failed applications directly. Moreover, to evaluate the performance of PJFNN fairly, we conducted experiments on both semisynthetic data and real-world data.

4.2 Evaluation on Person-Job Fit Prediction

Here, we will evaluate PJFNN by predicting whether an applicant fits a job, namely, the fitness between a given job posting and a resume.

Specifically, because PJFNN, which is for representation learning, does not apply to prediction problems directly, here we use the cosine similarities between representations of job postings and resumes learned by PJFNN as its prediction results.

To construct the dataset, we generate the same number of negative samples as that of positive samples. Along this line, we randomly selected 80% of the dataset as training data, another 10% for tuning the parameters, and the last 10% as a test set to validate the performance. To prove the effectiveness of PJFNN and our ideas about failed applications, we will demonstrate the performances of PJFNN and baselines, based on semisynthetic data (i.e., negative samples are manually generated) and real-world data (i.e., negative samples are randomly sampled from the failed application records). Besides, we also split our dataset according to time and validate the performance of our model in each year.

Baselines Methods. We first selected several classic classification methods as baselines, including *Logistic Regression (LR)*, *Decision Tree (DT)*, *Naive Bayes (NB)*, *Support Vector Machine (SVM)*, *Adaboost (Ada)*, *Random Forests (RF)*, *Gradient Boosting Decison Tree (GBDT)*, *Linear Discriminant Analysis (LDA)*, and *Quadratic Discriminant Analysis (QDA)*. For these baselines, it is unreasonable to directly use word vectors as input, which will lead to the curse of dimensionality. Thus, we treated the mean vector of all word vectors in a resume (job posting) as its latent vector, and then regard the latent vectors of a candidate's resume and the corresponding job posting together as the input of baseline methods. It is obvious that the quality of word representation has a large impact on the prediction performance. However, since both our model and these baselines are based on the same word representation, we think its quality cannot affect the validation of our model's effectiveness. Thus, we did not conduct experiments with different word embedding models and different dimensions of word representations.

Besides, because PJFNN maps the resume (job posting) onto a shared representation, it can also be treated as a dimensionality reduction method. Thus, we selected a dimensionality reduction method, *PLTM* (Mimno et al. 2009), as a baseline, which can discover topics aligned across multiple languages. Here, we treat resumes and job postings as different languages and represent both of them as distributions over a shared set of topics learned by PLTM. We calculated the cosine similarities between the distributions of job postings and resumes as its prediction results.

Evaluation Metrics. We evaluate the model performance by area under curve of ROC, or *AUC* for short, instead of other classic metrics for classification, such as *precision*, *recall*, and *F1*. This is because AUC can reflect models' performance within different boundary values between classes and thus is widely used in two-class classification problems.

4.2.1 *Evaluation on Semisynthetic Data.*

Here the positive samples are real successful applications in our dataset and the negative samples are manually generated by randomly selecting resumes and job postings. We demonstrate the AUC and significance test results in Table 1.

Table 1. AUC Performance of PFJNN and Baselines on Semisynthetic Data

	PJFNN	DT	LR	NB	SVM	LDA	QDA	Ada	RF	GBDT	PLTM
AUC	**0.85026**	0.65447	0.74757	0.67327	0.75241	0.76017	0.56764	0.71552	0.80104	0.83140	0.81568
Improve	–	+29.91%	+13.37%	+26.28%	+13.00%	+11.85%	+49.78%	+18.83%	+6.14%	+2.26%	+4.23%
P-value	–	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.001	<0.001

Table 2. AUC Performance of PFJNN and Baselines on Semisynthetic Data in Each Year

Method	2013			2014			2015			2016		
	AUC	Improve	P-Value	AUC	Improve	P-Value	AUC	Improve	P-Value	AUC	Improve	P-Value
PJFNN	**0.81891**	–	–	**0.86272**	–	–	**0.84486**	–	–	**0.81990**	–	–
DT	0.58573	+39.81%	<0.001	0.68980	+25.06%	<0.001	0.63743	+32.54%	<0.001	0.55047	+48.94%	<0.001
LR	0.73316	+11.69%	<0.001	0.75874	+13.70%	<0.001	0.72085	+17.20%	<0.001	0.74339	+10.29%	<0.001
NB	0.66495	+23.15%	<0.001	0.67823	+27.20%	<0.001	0.63483	+33.08%	<0.001	0.61375	+33.58%	<0.001
SVM	0.72413	+13.08%	<0.001	0.75348	+14.49%	<0.001	0.73071	+15.62%	<0.001	0.69660	+17.70%	<0.001
LDA	0.71475	+14.57%	<0.001	0.75627	+14.07%	<0.001	0.72081	+17.20%	<0.001	0.70405	+16.45%	<0.001
QDA	0.70876	+15.54%	<0.001	0.65188	+32.34%	<0.001	0.52678	+60.38%	<0.001	0.49262	+66.43%	<0.001
Ada	0.67320	+21.64%	<0.001	0.72488	+19.01%	<0.001	0.70941	+19.09%	<0.001	0.63677	+28.75%	<0.001
RF	0.69840	+17.25%	<0.001	0.80338	+7.38%	<0.001	0.79950	+5.67%	<0.001	0.73915	+10.92%	<0.001
GBDT	0.76122	+7.57%	<0.001	0.83079	+3.84%	0.004	0.82405	+2.52%	0.038	0.77446	+5.86%	<0.001
PLTM	0.79520	+2.98%	<0.001	0.81091	+6.38%	<0.001	0.80145	+5.41%	<0.001	0.71000	+15.47%	<0.001

Obviously, PJFNN has the best performance. Besides, PLTM, random forests, and GBDT also perform relatively well.

Besides, we split the dataset by time and train models separately on them for further evaluating our model within each year. The results are shown in Table 2. We can find that our model also outperforms all baselines in all of the years, which proves the effectiveness of PJFNN, again. Besides, the performances of all methods in 2014 and 2015 are obviously better than those in 2013 and 2016. We think it may be due to the lack of data. Actually, the numbers of successful applications in 2013 to 2016, which are 1,403, 5,492, 3,903, and 1,209, are highly correlated with the corresponding AUC results.

The corresponding AUC results for different job categories are listed in Table 3. Therein the performances of all methods are relatively poor in O. We think it may be because candidates' resumes cannot well reflect the skills required by jobs of O (e.g., marketing and customer service). After all, compared with other categories, where jobs are highly related to some professional skills (e.g., coding, program management, and UI design), jobs in O often do not have clear skill requirements and thus it is hard to find suitable candidates for them just by resumes.

4.2.2 Evaluation on Real-World Data. Here both positive samples and negative samples are real records in our dataset. We kept the proportion of positive samples to negative samples by randomly selecting the same number of failed applications as negative samples. Similarly, the performances on the entire dataset and on data of each year are shown in Table 5 and Table 6, respectively. Table 4 records the corresponding AUC results for different job categories on real-world data. Although the AUC results are much less than those on the semisynthetic data, PJFNN still generally outperforms these baselines. But what should be noted is that the performances of all methods are limited in data of 2015 and 2016. We think it can be attributed to the change of recruitment policy (e.g., partial

Table 3. AUC Performance of PFJNN and Baselines on Semisynthetic Data in Terms of Job Categories and Years

Year \ Job Category	T	P	U	O
2013	0.88267	0.87787	1.0	0.67677
2014	0.91797	0.88105	0.88194	0.75809
2015	0.90484	0.88596	0.73964	0.73606
2016	0.80120	0.90795	1.0	0.74122
overall	0.89811	0.89165	0.84131	0.76599

Table 4. AUC Performance of PFJNN and Baselines on Real-World Data in Terms of Job Categories and Years

Year \ Job Category	T	P	U	O
2013	0.79548	0.84303	0.85000	0.73190
2014	0.85283	0.79980	0.87878	0.68566
2015	0.75338	0.73597	0.83333	0.65537
2016	0.71839	0.67382	0.85454	0.60533
overall	0.77533	0.77313	0.74637	0.75367

Table 5. AUC Performance of PFJNN and Baselines on Real-World Data

	PJFNN	DT	LR	NB	SVM	LDA	QDA	Ada	RF	GBDT	PLTM
AUC	**0.75852**	0.59903	0.70009	0.60660	0.69541	0.69237	0.48609	0.67598	0.67926	0.71507	0.57500
Improve	–	+26.62%	+8.34%	+25.04%	+9.07%	+9.55%	+56.04%	+12.21%	+11.66%	+6.07%	+31.91%
P-value	–	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Table 6. AUC Performance of PFJNN and Baselines on Real-World Data in Each Year

Method	2013			2014			2015			2016		
	AUC	Improve	P-Value	AUC	Improve	P-Value	AUC	Improve	P-Value	AUC	Improve	P-Value
PJFNN	**0.77801**	–	–	**0.79549**	–	–	0.67099	–	–	0.67486	–	–
DT	0.65665	+18.48%	<0.001	0.58957	+34.92%	<0.001	0.59078	+13.57%	<0.001	0.62421	+8.11%	0.186
LR	0.73247	+6.21%	<0.001	0.67419	+17.99%	<0.001	0.66296	+1.21%	0.102	0.58629	+15.10%	<0.001
NB	0.69503	+11.93%	<0.001	0.63219	+25.83%	<0.001	0.56209	+19.37%	<0.001	0.58679	+15.00%	<0.001
SVM	0.72833	+6.82%	<0.001	0.71769	+10.84%	<0.001	0.65552	+2.35%	0.100	0.66165	+1.99%	0.377
LDA	0.65084	+19.53%	<0.001	0.68843	+15.55%	<0.001	**0.67860**	-1.1%	0.697	0.65315	+3.32%	0.981
QDA	0.60720	+28.13%	<0.001	0.53379	+49.02%	<0.001	0.44534	+50.66%	<0.001	0.48284	+39.76%	<0.001
Ada	0.59654	+30.42%	<0.001	0.65753	+20.98%	<0.001	0.63518	+5.63%	<0.001	0.64003	+5.44%	<0.001
RF	0.69726	+11.58%	<0.001	0.65889	+20.73%	<0.001	0.66118	+1.48%	0.127	0.65315	+3.32%	0.06
GBDT	0.69990	+11.16%	<0.001	0.71538	+11.19%	<0.001	0.65660	+2.19%	0.576	**0.67543**	-0.08%	0.581
PLTM	0.56283	+38.23%	<0.001	0.54371	+46.30%	<0.001	0.56283	+19.21%	<0.001	0.54371	+24.12%	<0.001

hiring freeze) beginning from October 2015, which results in that many excellent candidates cannot be enrolled although their qualifications fit the jobs well. Indeed, this situation further validates the reasonability of manually generating negative samples in some ways.

4.3 Evaluation on Joint Representation Learning

Since the direct goal of PJFNN is to learn a shared latent representation for both resumes and job postings, here we show the representation vectors of some resumes and job postings for validating the effectiveness of joint representation learning. Specifically, we randomly selected two resumes and two job postings from different job categories, and their vectors learned by PJFNN are shown in Figure 4, where the darker color means the higher value.

Job Posting #1. Considering the job posting contains the words "risk management," "data analysis," and "business insights," this job is clearly for recruiting people with risk management skills.

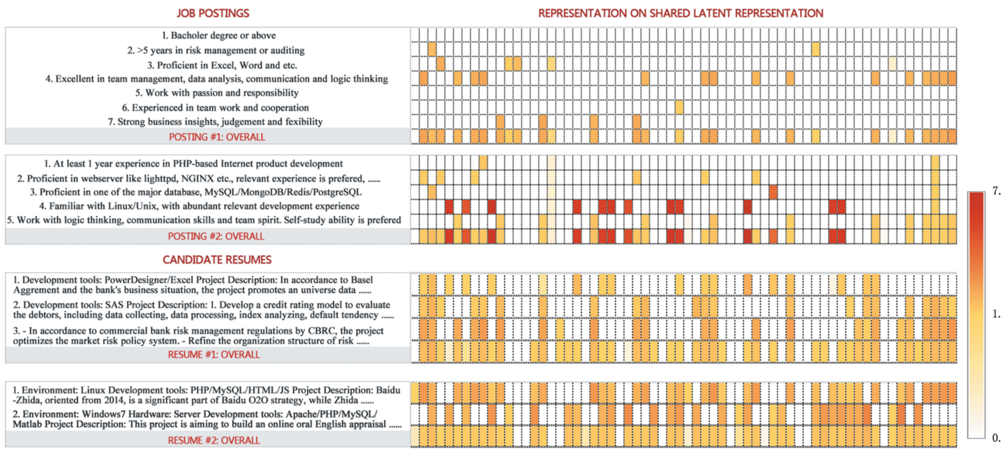


Fig. 4. The representations of some resumes and job postings learned by PJFNN, where the darker color means the higher value. The representation of each requirement item (work experience) and the overall representations of job postings (resumes) are all shown.

However, compared with requirement 2, which directly asks for ">5 years in risk management or auditing," requirements 3, 4, and 7 have more distinct vectors on the shared latent representation. By checking them, we can find that requirement 3, which asks for skills in "Excel" and "Work,", is obviously about the basic professional tools required for this job. Meanwhile, requirements 4 and 7, containing "data analysis," "logic thinking," "business insight," and "judgment," refer to the professional qualities for this job. In other words, these requirement items with distinct representations are highly relevant to the professional skills of this job. On the other hand, the representations of requirement 1 and requirement 5 are very vague. The first one refers to educational background and the fifth one is about working attitude. Obviously, in a high-tech company, both of them are very common requirements. Thus, we think their vague representations are also reasonable. Besides, we think the reason behind the vague representation of requirement 2 is that compared with the terminology "risk management," those skill-related words can appear in job postings more frequently and thus can been learned more distinctly. Actually, only about 60 out of 15,039 jobs contain "risk management" in our dataset.

Job Posting #2. Considering the keywords "PHP," "lighttpd," and "MySQL," this posting is for recruiting web development programmers. Actually, requirement 4 therein, which is about basic coding skills, has very distinct representations in this posting. It directly reflects the job content and the salience is reasonable. However, another technology-related requirement (requirement 2), which contains "PHP," "lighttpd," and "NGINX," is relatively vague. We think the reason behind this phenomenon is these web frameworks and specific skills are always being updated with rapid speed, and thus solid foundation of coding is more important. In fact, by checking a lot of requirements, we find that the web-related requirement items (e.g., for DJANGO and HTTP) often have more vague representations than those about basic coding skills, such as "C/C++," "Python," and "Algorithm."

Candidate Resume #1. It is obvious that all experiences of this candidate are about risk management in banks. For example, experience 2 is about developing "a credit rating model to evaluate debtors" and experience 3 is to "optimize the market risk policy system." Obviously this candidate has rich experiences in risk management and it is reasonable that her representation learned by PJFNN is close to that of job posting #1. Actually, cosine similarities between this resume and the

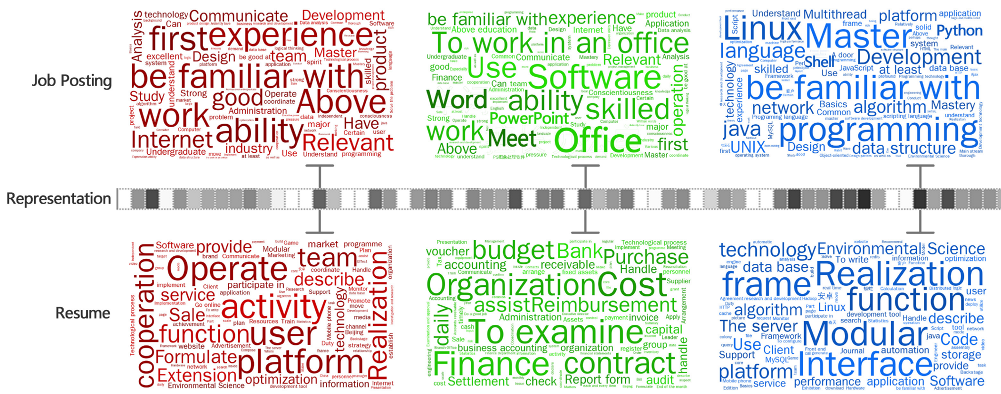


Fig. 5. The word clouds of three latent dimensions of representation learned by PJFNN, where the size of each keyword is proportional to its probabilities.

above two job postings are 0.89 and 0.12, respectively. This result further validates the effectiveness of our model.

Candidate Resume #2. The overall representation of this resume is relatively vague. Meanwhile the representation of experience 2 is distinct and is similar to job posting #2, which is about web development. Considering that its top-frequency words, containing "Apache," "PHP," and "online," are highly related to web technology, this similarity is reasonable. However, experience 1 is so vague that its latent vector is close to neither of the above job postings. Actually, the person described the project a lot but did not figure out her duty clearly. And PJFNN cannot extract much useful information. Thus, the similarities of this resume and both of the two jobs are not high (e.g., 0.42 for job posting #1 and 0.45 for job posting #2).

For further demonstrating the interpretability of PJFNN, we show some keywords for three randomly selected dimensions of the latent representation learned in Figure 5. Due to the structure of PJFNN, we cannot directly match keywords to latent dimensions. Thus, we first selected those resumes (job postings) whose values in the given dimensions are very high and extracted the high-frequency words from them as the keywords of this dimension. It is obvious that the first dimension is related to program management, where job postings often contain the keywords "communicate," "design," and "product." The frequencies of program-management-related words, such as "operate," "cooperation," and "team," are high in corresponding resumes. Meanwhile, the second dimension is more about administrative work, where job postings contain a lot of requirements for office software, such as "Office" and "Word." And the corresponding resumes, naturally containing "finance," "assist," and "organization," confirm this assumption. As for the third dimension, the high frequencies of coding-related keywords, such as "linux," "java," and "algorithm," clearly validate its relationship to IT development. This relationship is also proven from the perspective of resumes, where "modular," "interface," and "frame" have very high probabilities.

4.4 Empirical Studies

In this subsection, we will further empirically study some real-world Person-Job Fit cases based on the results of PJFNN.

We randomly selected some job requirements from job postings and work experiences from resumes and calculated their pairwise cosine similarities. The results are demonstrated in Table 7. We have highlighted the keywords in these work experience items manually. Requirement 1, which contains "C/C++ development" and "algorithms," is obviously about coding skills, and those work

Table 7. Some Matching Results between Job Requirements and Work Experiences, Where the Skill-Related Statements Are Highlighted for Facilitating Understanding

Concrete Job Requirement Items	Similarities	Concrete Work Experience Items
Requirement #1: Proficient in C/C++ development, familiar with relevant algorithms	0.99779	USD (Unified Storage Division) Senior Software Engineer. In charge of **file system development** of EMC middle-scale storage system. Have experience in many iterations. **Familiar with Unix file system, Common Block File System and 64-bit file system development.** Proficient in Volume . . .
	0.99689	Conducting Nachos OS kernel maintenance **on Linux with C++**. Primarily **be responsible for core modules like thread scheduling, file system and memory management**. Refined the algorithm of thread scheduling . . .
	0.40335	UI designer. Description: 1. Conducting **Graphic User Interface design** based on demands, for both mobile and TV client. 2. Established standards & manners for UI design. 3. Provided impression drawings . . .
	0.00056	Youjian Jiaoshi, **Intern Painting teacher** for kids between 4-10 years old. Giving lectures twice a week.
	0.00053	Supervisor of Shanghai Data Dept. Taking charge of **routine work**. Established the first off-site data collecting team. Be responsible for **data collecting outsourcing management**. Evaluating and testing data from multi-source.
	0.00053	Merchandising Manager. **Set up a company** with 2 partners in 2005, involved in the innovation and selling of measuring instrumentation chips. Gross income of 2006 was 2.8 million RMB, which climbed all the way to 25 million . . .
Requirement #2: Able to schedule and design products, familiar with product development procedure and documenting	0.86092	IBM, Senior Software Engineer. Worked in IBM CDL for more than 5 years. Be **responsible for WebSphere product** line and get PMP certification. **Abundant experience in product development, test and management** . . .
	0.82261	Be **responsible for RenRen mini-version** from the very beginning, re-directing to the "Social communication tools" market. Rebuilt the 20% popular functions (80% usage covered) of RenRen client following . . .
	0.31561	Lego is an editor-oriented system, consisting of video producing, topic customizing, page publishing and contract managing functions. This system includes **modules like jquery, jqueryui** and etc.
	0.31574	Joint with product lines of company, assisting to **solve problems with regard to user mode and kernel mode**. Make sure the innovated Linux-based products is commercially wide-used.
	0.07274	To **promote the scientific research and teaching combined**, and to culture the research abilities and team spirit of students, the school offers an innovative educational fieldwork project, relying on our research teams and . . .
	0.07267	TMS is a comprehensive intelligent traffic system. It integrates isometry devices and systems, in order to **resolve the isomerism of protocols and data**. I was **responsible for the resource management module**.
Requirement #3: With bachelor degree or above	0.57259	**Website Editor**. 1. Schedule the website pages structure, and plan for the contents. 2. Be responsible for mobile reading products. 3. Optimize product structure and functions with cross-dept communications
	0.26635	In charge of chess&card game operation and PC/mobile-client game-related translation for Facebook Japan. **Conducting data analysis** to enhance the DAU by 20%, and the quarter gross income by 100%.
	0.17677	This project is to break into the after-sale automobile service market, including preservation, repairment, rescue and second-hand sell, invested by CADIA. Be **responsible for the market investigation**.
	0.14005	Use **CNN to identify** the pattern of live eelworm. And help to enhance the robustness of real-time race.
	0.05386	USD (Unified Storage Division) Senior Software Engineer. In charge of **file system development** of EMC middle-scale storage system. Have experience in many iterations. **Familiar with Unix file system, Common Block File System and 64-bit file system development.** Proficient in Volume . . .
	0.52233	Be **responsible for RenRen mini-version** from the very beginning, re-directing to the "Social communication tools" market. Rebuilt the 20% popular functions (80% usage covered) of RenRen client following . . .

experience items with high similarities to it are also about coding. For example, the first work experience item has the highest similarity and it is about the file system development in Unix. The second one refers to "thread scheduling, file system, and memory management" in Linux. Besides, although the third experience, which is about UI/UE, is related to software development, it cannot unveil the coding skills of this candidate directly. Thus, we think its relatively low similarity is reasonable. Meanwhile, the other experiences, which are about education, data collection, or startups, have less relationship with coding and thus have low similarities.

Requirement 2, which contains "product development procedure" and "documenting," is obviously for program management. Thus, those work experience items about project management automatically have high similarities. In the first one, the candidate pointed out clearly that she was responsible for the product in IBM and had rich experience in product management. The second one also figured out directly the corresponding candidate's duty as a program manager in RenRen mini version. Thus, both of their similarities are high. Meanwhile, the similarities of those work experience items about coding, such as the third and fourth ones, are low accordingly. As for those IT-irrelevant experiences, such as the fifth and sixth ones, which are about education and resource management, respectively, there is no doubt that their similarities are close to zero.

However, not all of the job requirements can be modeled well in PJFNN. For example, requirement 3 is an education requirement, which widely appears in the job posting data. In the results, none of the work experience items have high similarity to it. As mentioned before, we think this phenomenon should be attributed to almost every job in our dataset that requires "With bachelor degree or above." Thus, PJFNN cannot learn a distinct representation for this requirement. In a word, we think that although PJFNN cannot learn good representations for all of the requirements, the latent vectors of most resumes and job postings learned by PJFNN are meaningful generally and can help to improve the effectiveness and efficiency of Person-Job Fit.

5 CONCLUSION

In this article, we proposed a novel end-to-end model based on CNN, namely, the Person-Job Fit Neural Network (PJFNN), for matching a talent qualification to the requirement of a job. To be specific, PJFNN is a bipartite neural network that can effectively learn the joint representation of Person-Job fitness from historical job applications; thus, it can project both job postings and candidates' resumes onto a shared latent representation. In particular, with the design of a hierarchical representation structure, PJFNN can not only estimate whether a candidate fits a job but also answer which requirements in the job posting are met by the candidate, through the measurement of distances between corresponding latent representations. Finally, we evaluated our model based on a large-scale real-world dataset collected from a high-tech company in China. The extensive experiments clearly validate the performance of our model in terms of Person-Job Fit prediction and demonstrate some interesting discoveries by visualizing the results obtained by PJFNN.

REFERENCES

Yu Cheng, Yusheng Xie, Zhengzhang Chen, Ankit Agrawal, Alok Choudhary, and Songtao Guo. 2013. Jobminer: A real-time system for mining job-related patterns from social media. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1450–1453.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–Decoder approaches. In *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8'14)*. 103.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics.* 1370–1380.

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*. ACM, 821–828.

Talent Acquisition Factbook. 2016. Benchmarking talent acquisition increasing spend cost per hire and time to fill. Retrieved from http://blog.bersin.com/benchmarking-talent-acquisition-increasing-spend-cost-per-hire-and-time-to-fill/. Accessed December 19, 2017.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual distributed representations without word alignment. In *Proceedings of International Conference on Learning Representations*.

John L. Holland. 1973. *Making Vocational Choices: A Theory of Careers*. Prentice Hall, Upper Saddle River, NJ.

Wenxing Hong, Siting Zheng, and Huan Wang. 2013. Dynamic user profile-based job recommender system. In *2013 8th International Conference on Computer Science & Education (ICCSE'13)*. IEEE, 1499–1503.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. 448–456.

Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom, Dimitri Kartsaklis, Nal Kalchbrenner, Mehrnoosh Sadrzadeh, Nal Kalchbrenner, Phil Blunsom, Nal Kalchbrenner, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 212–217.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1746–1751.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980* (2014).

Stanislas Lauly, Alex Boulanger, and Hugo Larochelle. 2014. Learning multilingual word representations using a bag-of-words autoencoder. *arXiv Preprint arXiv:1401.1803* (2014).

Jia Li, Dhruv Arya, Viet Ha-Thuc, and Shakti Sinha. 2016. How to get them a dream job?: Entity-aware features for personalized job search ranking. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 501–510.

Hao Lin, Hengshu Zhu, Yuan Zuo, Chen Zhu, Junjie Wu, and Hui Xiong. 2017. Collaborative company profiling: Insights from an employee's perspective. In *Proceedings of AAAI*. 1417–1423.

linkedIn Wikipedia. 2017. LinkedIn. https://en.wikipedia.org/wiki/LinkedIn. Accessed December 19, 2017.

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*. ACM, 963–966.

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*. IEEE.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv Preprint arXiv:1301.3781* (2013).

David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*. Association for Computational Linguistics, 880–889.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*. 807–814.

Ioannis Paparrizos, B. Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems*. ACM, 325–328.

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 *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 25–34.

Stephen P. Robbins. 2001. *Organizational Behavior* (14th ed.). Pearson Education India, Delhi.

Dazhong Shen, Hengshu Zhu, Chen Zhu, Tong Xu, Chao Ma, and Hui Xiong. 2018. A joint learning approach to intelligent job interview assessment. In *International Joint Conference on Artificial Intelligence*. 3542–3548.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*. 3104–3112.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28 (NIPS'15)*. 2773–2781.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 534–539.

Huang Xu, Zhiwen Yu, Jingyuan Yang, Hui Xiong, and Hengshu Zhu. 2016. Talent circle detection in job transition networks. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 655–664.

Tong Xu, Hengshu Zhu, Chen Zhu, Pan Li, and Hui Xiong. 2018. Measuring the popularity of job skills in recruitment market: A multi-criteria approach. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*. 2572–2579.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of CNN and RNN for natural language processing. *arXiv Preprint arXiv:1702.01923* (2017).

XianXing Zhang, Yitong Zhou, Yiming Ma, Bee-Chung Chen, Liang Zhang, and Deepak Agarwal. 2016. GLMix: Generalized linear mixed models for large-scale response prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 363–372.

Yingya Zhang, Cheng Yang, and Zhixiang Niu. 2015. A research of job recommendation system based on collaborative filtering. In *Proceedings of the 7th International Symposium on Computational Intelligence and Design*. 533–538.

Chen Zhu, Hengshu Zhu, Hui Xiong, Pengliang Ding, and Fang Xie. 2016. Recruitment market trend analysis with sequential latent variable models. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 383–392.

Received January 2018; revised April 2018; accepted June 2018