An Improved Deep Neural Network Model for Job Matching

Yu Deng
School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: herrdeng19830407@gmail.com

Hang Lei
School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: hlei@uestc.edu.cn

Xiaoyu Li
School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: xiaoyuuestc@uestc.edu.cn

Yiou Lin
School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: lyoshiwo@gmail.com

***Abstract* —Job matching which benefit job seekers, employees and employers is very important today. In this work, a deep neural network model is proposed to predict an employee's future career details, which includes position name, salary and company scale based on the online resume data. Like most NLP tasks, the input features are multi-field, non-sparse, discrete and categorical, while their dependencies are mostly unknown. Previous works were mostly focused on engineering, which resulted in a large feature space and heavy computation. To solve this task, we use embedding layers to explore feature interactions and merge two automatically learned features extracted from the resumes. Experimental results on over 70,000 real-word online resumes show that our model outperforms shallow models, like SVM and Random Forests, in effectiveness and accuracy.**

Keywords-deep neural network; natural language processing; word embedding; job matching; recommender system

I. INTRODUCTION

As the economic environment persisted, a large number of financial institutions and business companies went bankrupt. Even in the richest countries, knowledge workers have to worry about losing their full-time, well-paid jobs, and it is difficult to find similar jobs elsewhere. Also, it is very time-consuming and laborious to identify candidates with the right potential traits. Improving the quality of recruitment would assign the right personnel to the right position, which could make the employees to achieve greater work performances and significantly reduce the employee training cost.

At the same time, the Internet-based recruitment platform has become the main recruitment channel for most businesses. It improves the efficiency of the recruitment, saves the advertising cost and releases the enterprise from information overload. Traditional information retrieval techniques have been used for job matching, such as Boolean retrieval methods, but they are not suitable for this task. As a result, a large number of candidates missed the opportunity to recruit [1]. An effective job matching system can help job seekers to get recruitment opportunities more easily, and it will also reduce the work of employers, because the system provides appropriate items in line with their interests and qualifications.

Job matching, however, is full of challenging because knowledge of individual interests, academic, professional and vocational qualifications are often difficult to identify. More importantly, without prior knowledge of the recruiting market, the researchers could not have an in-depth understanding of job seekers. These job seekers are characterized by specific individual attributes and different from each other. Therefore, the main purpose of job matching is trying to engage the job searchers who are well suited for specific vacancies. It is very difficult to achieve this in the job market, which makes our work urgent and meaningful.

Most decision-making problems or choice problems in the real world fall into the multi-attribute evaluation category. As a classification problem, Job matching is to tap the current employability of job seekers, according to their previous career history. To solve this task, we design an improved deep neural network model which contains embedding layer to explore deep feature interactions. Meanwhile two kinds of features extracted from the resumes are merged in the model. We evaluate our approach on an online resume dataset, which contains 70,000 real-word resumes. The main contributions of our work can be summarized as follows:

- After analyzing the real-word online resume dataset, we propose two kinds of features to handle job matching task. One is categorical feature which is manual

978-1-5386-6987-7/18/$31.00 ©2018 IEEE

designed from resume structure, and another is context feature which is automatically learned by word2vec.

- A deep neural network model with embedding layer is proposed, which can explore deep feature interactions. we improve its performance by merging two kinds of features as input.
- Experimental results indicate that our approach can improve the effectiveness and accuracy compared with several other baselines, such as SVM, RF and XGB.

The rest of the paper is organized as follows. In Section II, we survey the related literatures to give an overview of the research background. In Section III, three typical machine learning models are briefly discussed. In Section IV, we introduce our methodology-including the dataset description, models, feature extraction methods and metrics methods. In Sections V, experimental procedures are introduced and the empirical results are analyzed. Finally, the paper is concluded in Section VI with our future work.

II. Related Work

Recently, intelligent systems based on flexible and customizable solutions have been widely used in the areas such as NLP, robotics and medical diagnosis. Meanwhile, the application of these solutions to online recruitment systems has become the focus of many studies, which enhance the competitive advantage.

Job matching system is generally, a kind of recommender system. Resnick and Varian first pointed out that recommendations were provided as inputs in a typical recommender system, which were aggregated and directed to appropriate recipients [2]. In general, recommender systems are applied in various domains such as books, digital products, movies, TV programs, and web sites. Nowadays, recommender systems help users to find contents, products, or services by aggregating and analyzing suggestions and behaviors from other users. A detailed survey paper [3] provided researchers with the state-of-art knowledge on recommender system including recommendation methods, real-world applications and application platforms.

Many researches have been conducted to discuss different recommender strategies related to the recruiting problem. In one of the recent studies, Park presented a match-making system that could adaptively adjust the recommendation model reflecting the user's implicit and explicit preferences [4]. When the system provided recommendations for new users based on their assigned explicit preference weights, it could automatically adjust the weight of each attribute by analyzing their previous behaviors using logistic regression.

Zhu proposed a framework for calculating asymmetric conceptual skill similarity based on weighted-path-counting, which was validated in a use case of programming job matching [5]. Sisay Chala discussed the approaches of vacancy extraction and representation based on required and desired criteria, and then showed how plain keyword-based vacancy-to-jobseeker matching might result in improper matching [6]. Alfonso Arpaia discussed the main features of job matching in the EU after the 2008-2009 worldwide crisis [7], while Steve McDonald examined how institutional arrangements affected network-based job finding behaviors in the United States and Germany [8]. In recent years, more specific occupational and labor market information is gathered and considered by researchers to ultimately achieve much better career decisions and job placements [9]. Hyder and Chen gathered the information from job seekers and prospective employers, then provided matching job results based on common parameters between job vacancies and prospective employees [10]. Guo presented a personalized job-resume matching system, which offers a novel statistical similarity index for ranking relevance between candidate resumes and a database of available jobs [11].

There are several disadvantages in the above works and systems. Firstly, specific occupational and labor market information with prior expert knowledge is difficult to collect. Secondly, artificial designed strategies and feature extraction methods are usually subjective and not robust. At last, pattern matching and linear models, which have advantages of easy implementation, show relatively low performance in learning non-trivial patterns to catch the interactions between the multi-field categorical features.

Despite the effectiveness of those methods, it is still challenging to tackle these job matching issues on a big online resume dataset. Therefore, we are motivated to design an improved deep neural network model to extract the interactional features, hidden structures and intrinsic patterns, which is quite different from traditional machine learning models.

III. Machine Learning Models

In this section, three typical machine learning models which are widely used in recommender systems will be briefly discussed. Meanwhile we consider them as test baseline for measuring the DNN model proposed in this work.

A. SVM Model

The Support Vector Machine (SVM) model has been applied in recommender system for many years which selects the hyperp-lanes to separate two classes of data, so that the distance between them is as large as possible [12]. LinearSVC is an implementation of SVM for the case of a linear kernel. The hyperplanes can be described by the equations $\mathbf{w} \cdot \mathbf{x}\text{-}\mathrm{b} = +1$ and $\mathbf{w} \cdot \mathbf{x}\text{-}\mathrm{b} = \text{-}1$ where $\mathbf{W}$ is the normal vector to the hyperplanes.

Geometrically, the distance between these two hyperplanes is $2/\|\mathbf{w}\|$. Since minimizing $2/\|\mathbf{w}\|$ is equivalent to minimizing $\|\mathbf{w}\|^2/2$, the optimization problem is finally converted to equations as follows.

$$\min_{\mathbf{w},\mathbf{b}} \ \frac{1}{2}\|\mathbf{w}\|^2 \tag{3}$$

$$s.t. \ \ y_i(\mathbf{w} \cdot x_i + \mathrm{b}) - 1 \geq 0, \ \ i = 1, 2, ..., N \tag{4}$$

LinearSVC supports both dense and sparse input. Meanwhile, the multiclass support is handled according to a one-vs-the-rest scheme.

B. *Tree-Based Model*

Random Forests (RF) and Extreme Gradient Boosting (XGB) are typical tree-based models. Both are a set of classification and regression trees. Compared to XGB, RF is much easier to understand. At each candidate split in the learning process, a random subset of the features is selected by RF called "feature bagging" to avoid trees in RF becoming correlated. Unlike the traditional optimization problem where the gradient can be taken, training the parameters of XGB is to fix those trees which have been learned, and to add one new tree at a time. Thus, at step t, for the sample x_i and tree f_t, the predication is as follows

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{3}$$

Then the MSE loss function will become the following form with Ω as regularization.

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n} (\hat{y}_i^{(t)}, y_i) + \sum_{i=1}^{t} \Omega(f_i) \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \\ &= \sum_{i=1}^{n} (2(y_i - \hat{y}_i^{(t-1)}) f_t(x_i) + f_t(x_i)^2) \\ &\quad + \Omega(f_t) + constant \end{aligned} \tag{4}$$

IV. METHODOLOGY

This section presents a detailed description of the methodology used in our work, and a process illustration is given in Figure 1. After analyzing the characteristics of online resume dataset, we propose methods to extract categorical features and context features, and then build a basic deep neural network model based on categorical features. Finally, we add context features in the model for improvement.

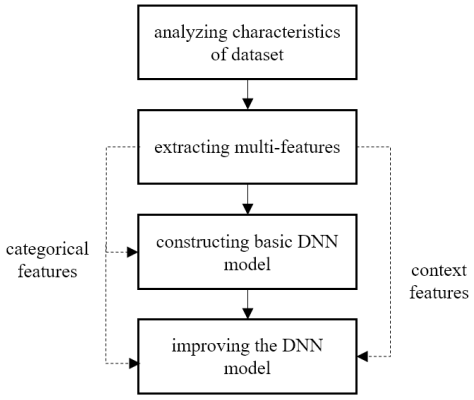

Figure 1. the process illustration of methodology.

A. *Dataset description*

The original dataset includes 70,000 Chinese resumes provided by DataCastle[1] with 31,924 different positions. Each resume is made up of user's age, gender, major profession, educational level and employment history，as shown in Figure 2.

Through a detailed statistical analysis of the dataset, we found that there are 18,732 different positions with long-tail distribution in the dataset, as illustrated in Figure.3. Every single resume has a typical sequential structure, which is similar to the text handled in NLP task. Therefore, the meaning of a particular token in a given resume is decided by both its definition and its context.

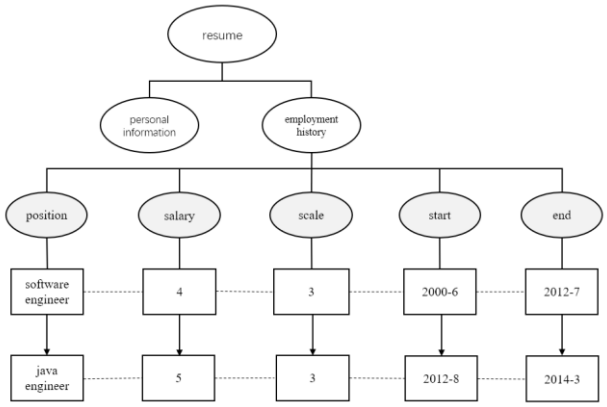

Figure 2. the structure of a resume with previous employment history.

After cleaning and filtering, 47,346 resumes whose last job belong to a particular prediction list of most frequent 32 positions have been gained, and our problem is redefined to predict a job seeker's current position (last position in the resume) through employment history and personal information.

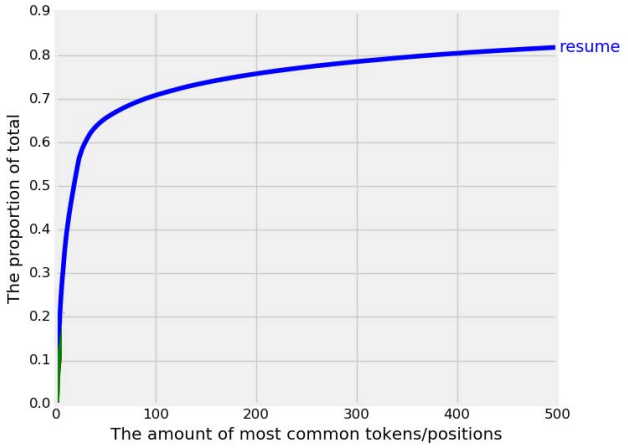

Figure 3. most frequent positions possess the proportion in resume data.

B. *Feature Extraction Methods*

1) *Categorical features*

We found that there were common points between POS (part-of- speech) tagging and resume analysis in feature

[1] http://www.pkbigdata.com

extraction. For decades, POS tagging has been the basic and classic NLP task and drawn lots of research attention. Many researchers have proposed different methods to extract effective features in POS tagging, such as spelling features, lexical features, n-gram features, word embedding and cluster-based features [13-14]. we will draw lessons from POS tagging and discuss the feature extraction. In this work, some multi-field categorical features are designed to play the roles of spelling features, lexical features and n-gram features.

The design of categorical features is common to our researchers and we all think it is one of the key factors to construct successful machine learning applications. These categorical features can present and describe the structure and information inherent in the original data.

We extract the following main features for a given resume.

- Personal information including gender, age, major and degree.
- The age when first employed.
- The title of previous position, such as manager, director and chairman
- The previous position encoded using the index.
- The previous salaries and scales
- The previous working periods

Similar to BOW (bag of words) model, we finally turned each resume to a 70-dimensional vector, which was represented by the classification features.

2) Context features

Although feature engineering plays a very important role in machine learning, a simple increase in the number of artificial features does not break through the limit of predictive performance. For exploring deep feature interactions, word embedding features are used in this work as context features for practical purposes.

Word embedding is a family of NLP techniques aiming at mapping semantic meaning into a geometric space. It captures part of the semantic relationship between associated words by calculating a numeric vector to every word in a dictionary. Collobert verified that word embedding plays a vital role to improve POS tagging performance [15].

A model to generate this mapping using language model is as follows. We use a random vector marked as X_t for the t-th word, and we suppose that whether X_t appears or not is only decided by the language model f with parameters λ and its context, X_{t-1} and X_{t+1}. We can then define the probability of X_t as

$$p(\mathbf{x}_t \mid context_t) = f(context_t, \lambda) = f(\mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \lambda) \quad (1)$$

The training process is to identify $\mathbf{x}$ and λ such that

$$\mathbf{x}, \lambda = \arg\max_{\mathbf{x},\lambda} \prod_{t=2}^{N-1} f(context_t, \lambda) \quad (2)$$

A popular implementation is called word2vec and it was created by a team of researchers led by Tomas Mikolov at Google [16-17]. Word2vec is a method to obtain distributed representations for a word by using neural networks with one hidden layer. It learns neural network models from large texts by solving a pseudo-task to predict a word from surrounding words in the text. The word weights between the input layer and the hidden layer are extracted from the network as the distributed representation.

We use word2vec to reconstruct linguistic contexts of features and made each position, degree, salary, major and age correspond to a 10-dimensional embedding vector. Here are the procedures how we train the word embedding：

Step1 Convert all resumes in the dataset into sequences of token in the same format.

Step2 Truncate the sequences to a maximum length of 16 tokens without the final target job details, including 4 tokens for personal information (major, degree, age, gender), the 12 tokens represent a maximum amount of 3 jobs (each job has position, salary, company size and working seasons).

Step3 Calculate word embedding for each token using word2vec, and each token corresponds to a 10-dimensional embedding vector.

A visualization of correlative word embedding using t-sne is shown in Figure 4.

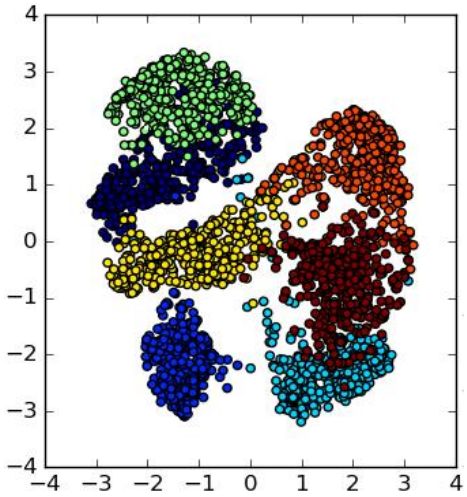

Figure 4. t-sne visualization of correlative word embeddings for 7 selected positions.

C. DNN Model Architecture

In this section, the improved DDN model proposed in this work will be discussed in detail. First, we will construct a basic model with four layers, then pre-trained word embedding features will be merged into the model, which finally become a multi-input neural network.

1) Basic DNN model

There are four layers in our basic DNN model, from bottom to top, they are embedding layer, flatten layer, hidden layer and SoftMax layer. The network structure is shown in Figure.5. The following details will be introduced with a down-top description.

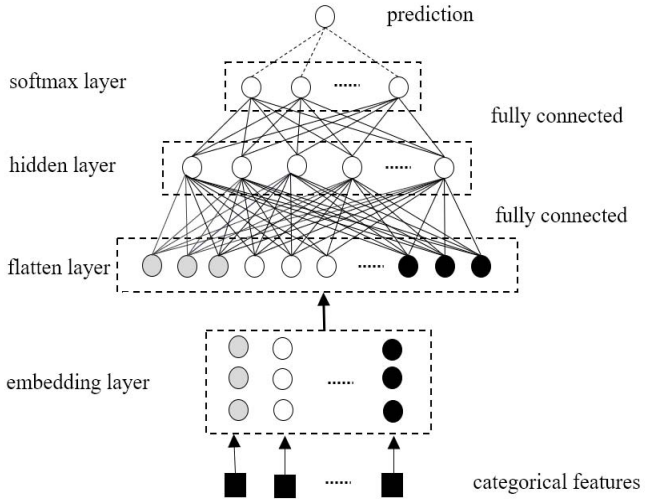

Figure 5. the structure of basic DNN model.

The embedding layer as input is at the bottom of the basic structure, which is fed sequences of categorical features. As mentioned, we set an embedding layer to turn positive integers (indexes of categorical features) into dense vectors of fixed size. The dense vectors will be continuously trained in end-to-end supervised learning, so that the vectors that share common contexts in the corpus will be finally located near one another in the vector space. In addition, word embedding is a more effective way than traditional sparse coding method, such as one-hot representation.

The flatten layer will combine the components of the two-dimensional vector matrix produced by the embedding layer into one dimensional vector in order, which will be took as input tensor by next layer.

Basic structure has just one hidden layer, which is fully connected. It enhances the ability of mining hidden nonlinear characteristics for the model.

SoftMax layer is at the top of the basic structure, which plays a role as a multi-category output layer to predict the training labels. The number of nodes in this layer will be set the same as the predication list. SoftMax is widely used to solve the multi classification problem in machine learning, because it is simple and efficient in computing. It normalizes multiple inputs and outputs the probability of the corresponding class. Mathematically, the SoftMax function to calculate the final score is as follow, where K is the number of a prediction list.

$$\text{soft}\max(w_j) = \frac{\exp(w_j)}{\sum_{k=1}^{K}\exp(w_k)} \qquad (5)$$

2) Improved DNN model

After constructing the basic structure, we use context features (pre-trained embedding matrix) to improve the DNN model, which finally become a multi-input neural network. An illustration of our improved DNN model is given in Figure 6.

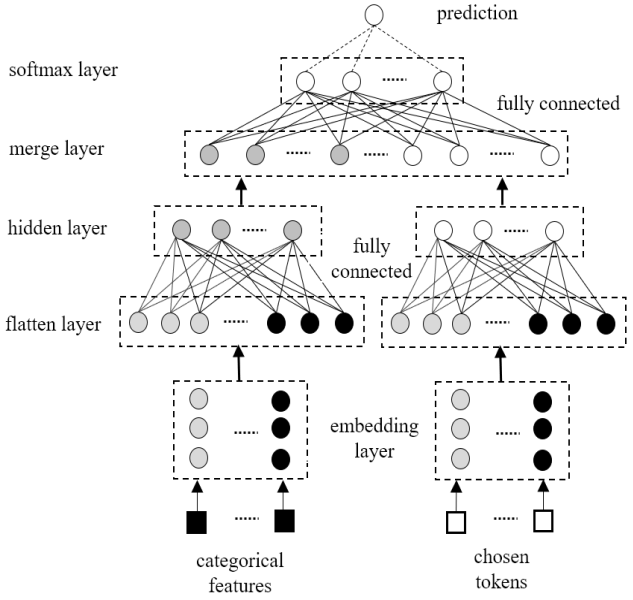

Figure 6. the structure of improved DNN model.

The pre-trained word embedding matrix has been introduced in section IV.b. The "embedding matrix" is produced by word2vec, which contains the embedded vectors for each carefully chosen token. Compared to embedding layer, word2vec adopts a quite different way to generate word vector, which is unsupervised learning. It is completely dependent on the context itself, not other aspects. So the "embedding matrix" represents a wider semantic vector space, and it will help our model to explore more deep feature interactions

There are two branches in the improved model. Each branch has the same structure as basic model. It is worth of mentioning that the details of the embedding layer in the right branch (bottom right) are different. It is fed the sequences of chosen tokens, and the pre-trained embedding matrix is used as its initial weights. After loading the embedding matrix into the embedding layer, the embedding vectors will be continuously updated in training.

Finally, we add a merge layer under the SoftMax layer, which concatenate the different tensors output from two branches.

V. EXPERIMENT

A. Experiment Setup

The main experiment purposed in this work is to predict future career conditions-including position, salary and

company scale. We select 47,346 resumes whose last jobs belong to a particular prediction list of most frequent 32 positions. A total of 67% resumes are randomly selected as training data and the rest are selected as test data. All the models share the common features introduced in section IV.b. Since SVM only handles numeric and binary feature, after one-hot encoding, categorical features are transformed into binary feature. Meanwhile, grid search and cv-test for best parameters are used for both tree-based models. The error rate of XGB model with different parameters is shown in Figure 7.

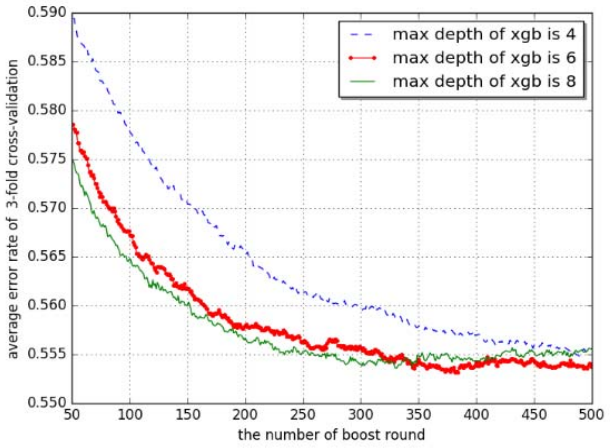

Figure 7. the error rate of position prediction evaluated by XGB with different parameters.

Finally, we select the micro (the proportion of correct samples in all samples) and macro precisions (the average correct rate for all different labels) as our metrics methods.

B. *Hyper Parameter Tuning*

We use Keras to implement our neural network models, and TensorFlow is backend. The number of parameters needed to be trained in the neural network is huge. For example, over the resume corpus, there are 70,014 different categorical features, causing the generation of 2,100,420 parameters in the embedding layer of the basic DNN model, as show in Table I.

TABLE I. THE AMOUNT OF PARAMETERS IN DIFFERENT LAYERS OF ORIGINAL DNN MODE

Layer	Output shape	Parameter
Embedding	(None,70,30)	2100420
Flatten	(None,2100)	0
Hidden	(None,256)	537856
SoftMax	(None,32)	8224
Total		2646500

Hyper parameter tuning is also called fine-tuning, which is the key to optimizing the performance of the model. The following details show how we tune hyper parameters in our improved DNN model.

We use RMSProp optimizer to learn our parameters. Regarding selecting the number of training epochs, we use early stopping when the validation error increases as shown in Figure 8.

For the activation functions in both models on the hidden layers, we try ReLU function, sigmoid function and tanh function, and we find that the result of ReLU function is optimal. At last we try different number of hidden units and choose the one with optimal performance on the validation dataset.

We randomly drop 50% units from the neural network during training. It prevents units from co-adapting too much and approximates the effect of averaging the predictions.

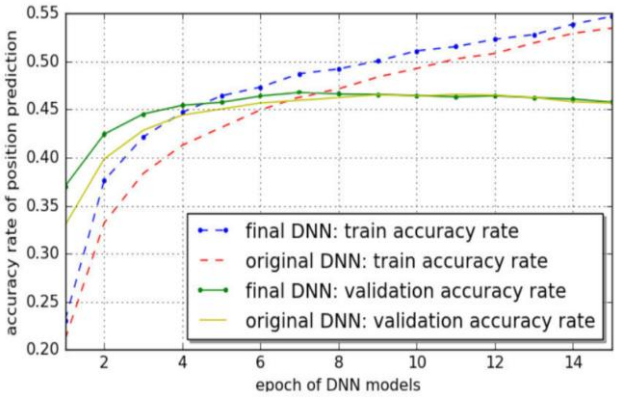

Figure 8. accuracy rate by different training epochs.

C. *Performance Comparison*

We report empirical results of four different models as shown in table II-III. Among three traditional models, SVM is the weakest baseline for all the three labels and RF performs close to XGB. Comparing with these models, our improved DNN model shows great competitiveness and wins in all metrics methods.

TABLE II. MICRO PRECISION OF DIFFERENT MODELS

Label	SVM	RF	XGB	DNN
Feature	One-hot	Manual	Manual	Embedding
Scale	0.323	0.377	0.385	**0.401**
Salary	0.434	0.508	0.510	**0.522**
Position	0.393	0.420	0.453	**0.472**

TABLE III. MACRO PRECISION OF DIFFERENT MODELS

Label	SVM	RF	XGB	DNN
Feature	One-hot	Manual	Manual	Embedding
Scale	0.305	0.372	0.383	**0.411**
Salary	0.426	0.532	0.532	**0.545**
Position	0.307	0.403	0.407	**0.424**

It is interesting that our improved model for job matching has less dependence on unsupervised word embedding than POS task does. As shown in Figure 8, the unsupervised word embedding helps model reach convergence faster, but it does not improve the precision significantly. We believe that there are two main reasons for this phenomenon. First, despite the features designed in this paper are very appropriate, the resume dataset is still too small for effective training. Secondly, the resume data has a serious long tail problem with weak context relationship, which results in training low frequency token ineffectively.

VI. Conclusion and Future Work

We have constructed a detailed DNN solution for career prediction. Compared to different machine learning methods, the experiments demonstrate that our improved DNN model is more effective and accurate. In the future, with more information to be snatched from website we will try to extend our work in multiple languages. Meanwhile, we will extend further study using additional knowledge from unlabeled web text and personalized information, such as location, professional skills and description of requirements from both job seekers and employers.

Acknowledgment

The financial support for this work is provided by the National Natural Science Foundation of China, No. 61502082.

References

[1] Al-Otaibi S. T., Ykhlef M. (2012). A survey of job recommender systems. International Journal of Physical Sciences, 7(29), 5127-5142.

[2] Resnick P., & Varian H. R. (1997). Recommender systems. Communications of the ACM, 40(3), 56-58.

[3] Lu J., Wu D., Mao M., Wang W., Zhang G. (2015). Recommender system application developments: a survey. Decision Support Systems, 74, 12-32.

[4] Park, Y. J. (2013). An adaptive match-making system reflecting the explicit and implicit preferences of users. Expert Systems with Applications, 40(4), 1196–1204.

[5] Bo Zhu, Xin Li, Jesus Bobadilla Sancho. (2017). A Novel Asymmetric Semantic Similarity Measurement for Semantic Job Matching. 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), 152-157. Shenzhen, China.

[6] S Chala, S Harrison , M Fathi. (2017). Knowledge extraction from online vacancies for effective job matching. 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 1-4. Windsor, Canada

[7] Arpaia A., Kiss A., Turrini A. (2014). Is unemployment structural or cyclical : Main features of job matching in the EU after the crisis. Economic papers, 527, 1-60.

[8] McDonald S., Benton R. A.,Warner D. F. (2012). Dual embeddedness: Informal job matching and labor market institutions in the United States and Germany. Social forces, sos069.

[9] JA Johnston, KL Buescher, MJ Heppner.(2014). Computerized career information and guidance systems: Caveat emptor. Journal of Counseling & Development, 2014 , 67 (1) :39-41.

[10] Hyder A, Chen C. (2010). Intelligent job matching system and method including preference ranking. U.S. Patent No. 7,720,791. Washington, DC

[11] Guo S., Alamudun F., Hammond T. (2016). ResuMatcher: a personalized resume-job matching system. Expert Systems with Applications, 60, 169-182.

[12] Burges CJC (1998) A tutorial on support vector machine for pattern recognition. Data Min Knowl Discov 2(2):121–167

[13] Huang Z., Xu W., Yu K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

[14] O Owoputi, B O'Connor, C Dyer, K Gimpel, N Schneider. (2013). Improved part-of-speech tagging for online conversational text with word clusters. Proceedings of NAACL-HLT 2013, 380–390. Atlanta, Georgia

[15] Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.

[16] Mikolov T, Chen K, Corrado G, Dean J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[17] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111–3119.