Appendix A: Proof of the Variational Inference Process in Section 3.3

Before showing the proof of each term in the Evidence Lower Bound (ELBO) of $\mathcal{L}_0$, we first summarize some formulas which will be used in related proofs.

$$\frac{\partial(\log[\mathbf{B}(X)])}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\sum_i^I \log[\Gamma(x_i)] - \log\left[\Gamma\sum_i^I x_i\right]\right) = \mathbf{\Psi}(x_i) - \mathbf{\Psi}(\sum_i x_i), \tag{A.1}$$

where $X = x_1, x_2, ..., x_I$.

If $\Theta \sim Dirichlet(\alpha_1, \alpha_2, ..., \alpha_i, ..., \alpha_I)$, where $\mathbf{\Theta}$ is a simplex, then we have

$$E[\log(\theta_i)] = \frac{\partial \log(\mathbf{B}(\alpha))}{\partial \alpha} = \mathbf{\Psi}(\alpha_i) - \mathbf{\Psi}(\sum_k^K \alpha_k). \tag{A.2}$$

$$f = \sum_i^I \sum_{g=i+1}^I A_i Z_g = \sum_i^I \sum_g^{i-1} A_g Z_i. \tag{A.3}$$

A.1. Proof of Eq. (11)

Proof. Given that

$$\begin{aligned} E_q[c_{kl}] &= \widetilde{\mu}_{c_{kl}}, \\ E_q[c_{kl}^2] &= \widetilde{\mu}_{c_{kl}}^2 + \widetilde{\lambda}_{c_{kl}}^{-1}, \end{aligned} \tag{A.4}$$

we have

$$\begin{aligned} \mathcal{L}_1 &= -\frac{1}{2} E_q[h_{ij}(s_{ij} - t_i^T c_k)^2] \\ &= -\frac{1}{2} h_{ij}\left\{s_{ij}^2 - 2 s_{ij} t_i^T E_q[c_k] + E_q[c_k^T t_i t_i^T c_k]\right\} \\ &= -\frac{1}{2} h_{ij}[s_{ij}^2 - 2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + (t_i^T \widetilde{\mu}_{c_k})^2 + \sum_l^L \widetilde{\lambda}_{c_{k,l}}^{-1} t_{il}^2] \\ &= -\frac{1}{2} h_{ij}\left\{s_{ij}^2 - 2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + t_i^T [\widetilde{\mu}_{c_k} \widetilde{\mu}_{c_k}^T + \mathbf{\Lambda}(\widetilde{\lambda}_{c_k}^{-1})] t_i\right\} \\ &= -\frac{1}{2} h_{ij}\left\{s_{ij}^2 - 2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + t_i^T \rho_k t_i\right\}. \end{aligned} \tag{A.5}$$

■

A.2. Proof of Eq. (15)

Proof.

$$\begin{aligned} E_q[\log(P(x_{jd}|z_j, \psi_{kd}))] &= E_q[\log\left(\prod_k^K P(x_{jd}|\psi_{kd})^{1[z_j=k]}\right)] \\ &= E_q[\sum_k^K 1[z_j = k] \cdot \log P(x_{jd}|\psi_{kd})] \\ &= \sum_k^K \left(E_q[1[z_j = k]] \cdot E_q[\log \psi_{kd, x_{jd}}]\right) \\ &= \sum_k^K \widetilde{z}_{jk} E_q[\log \psi_{kd, x_{jd}}]. \end{aligned} \tag{A.6}$$

■

A.3. Proof of Eq. (16)

Proof.

$$\begin{aligned} E_q[\log(P(\psi_{kd}|\gamma))] &= E_q[\log \frac{1}{\mathbf{B}(\gamma)} \prod_m^M \psi_{kdm}^{\gamma-1}] \\ &= \sum_m^M (\gamma-1) E_q[\psi_{kdm}] - \log \mathbf{B}(\gamma). \end{aligned} \tag{A.7}$$

■

A.4. Proof of Eq. (18)

Proof.

$$\begin{aligned} E_q[\log(q(z_j|\widetilde{z}_j))] &= E_q[\log \sum_k^K \widetilde{z}_{jk}^{1[z_j=k]}] \\ &= \sum_k^K E_q[1[z_j=k]] \log(\widetilde{z}_{jk}) \\ &= \sum_k^K \widetilde{z}_{jk} \log(\widetilde{z}_{jk}). \end{aligned} \tag{A.8}$$

■

Appendix B: Proof of Updating Formulas in Section 3.4

This appendix shows the mathematical proof of updating strategies in the optimization of our model.

B.1. Proof of Eq. (21)

Proof. We first extract all the terms contain $\widetilde{\theta}_k$ and get

$$\mathcal{L}\left(\widetilde{\theta}_k\right) = \left(\sum_j^J \widetilde{z}_{jk} - \widetilde{\theta}_{k,1} + 1\right) E_q[\log(\theta_k)] + \left(\sum_j^J \sum_{g=k+1}^K \widetilde{z}_{jg} - \widetilde{\theta}_{k,2} + \beta\right) E_q[\log(1-\theta_k)] + \log\left(\mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2}\right). \tag{B.1}$$

To present the proof process more concisely, we substitute some terms with simple notations. They are:

$$\begin{gathered} E_q[\log(\theta_k)] = f_1, \quad E_q[\log(1-\theta_k)] = f_2, \\ \frac{\partial f_1}{\partial \widetilde{\theta}_{k,1}} = f_{11}, \quad \frac{\partial f_1}{\partial \widetilde{\theta}_{k,2}} = f_{12}, \quad \frac{\partial f_2}{\partial \widetilde{\theta}_{k,1}} = f_{21}, \quad \frac{\partial f_2}{\partial \widetilde{\theta}_{k,2}} = f_{22}. \end{gathered} \tag{B.2}$$

And we can get

$$\frac{\partial \mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2})}{\partial \widetilde{\theta}_{k,1}} = f_1, \quad \frac{\partial \mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2})}{\partial \widetilde{\theta}_{k,2}} = f_2. \tag{B.3}$$

Now, we can calculate the deviations of $\mathcal{L}(\widetilde{\theta}_k)$ with the above substitutional notations. By setting the deviations to zeros, we get

$$\begin{aligned} \left(\sum_j^J \widetilde{z}_{jk} - \widetilde{\theta}_{k,1} + 1\right) f_{11} + \left(\sum_j^J \sum_{g=k+1}^K \widetilde{z}_{jg} - \widetilde{\theta}_{k,2} + \beta\right) f_{21} = 0, \\ \left(\sum_j^J \widetilde{z}_{jk} - \widetilde{\theta}_{k,1} + 1\right) f_{12} + \left(\sum_j^J \sum_{g=k+1}^K \widetilde{z}_{jg} - \widetilde{\theta}_{k,2} + \beta\right) f_{22} = 0. \end{aligned} \tag{B.4}$$

After solving the equations above, we can get the updating formulas as follows.

$$\begin{aligned} \widetilde{\theta}_{k,1} &= 1 + \sum_{j}^{J} \widetilde{z}_{jk} \\ \widetilde{\theta}_{k,2} &= \beta + \sum_{j}^{J} \sum_{g=k+1}^{K} \widetilde{z}_{jg} \end{aligned} \tag{B.5}$$

■

B.2. Proof of Eq. (22)

Proof. We first extract all the terms containing $\widetilde{\mu}_{c_k}$ and $\widetilde{\lambda}_{c_k}$:

$$\begin{aligned} \mathcal{L}(\widetilde{\mu}_{c_k}) &= \sum_{i}^{I} \sum_{j}^{J} \widetilde{z}_{jk} \left(-\frac{h_{ij}}{2}\right) \left[-2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + (t_i^T \widetilde{\mu}_{c_k})^2\right] - \frac{\lambda_c}{2} \widetilde{\mu}_{c_k}^T \widetilde{\mu}_{c_k}, \\ \mathcal{L}(\widetilde{\lambda}_{c_k}) &= \sum_{i}^{I} \sum_{j}^{J} \widetilde{z}_{jk} \left(-\frac{h_{ij}}{2}\right) \sum_{l}^{L} \widetilde{\lambda}_{c_{kl}}^{-1} t_{il}^2 - \frac{\lambda_c}{2} \sum_{l}^{L} \widetilde{\lambda}_{c_{kl}}^{-1} - \sum_{l}^{L} \frac{1}{2} \log\left(\widetilde{\lambda}_{c_{kl}}\right). \end{aligned} \tag{B.6}$$

Then, we calculate the deviations of Eq. (B.6):

$$\begin{aligned} \frac{\partial}{\partial \widetilde{\mu}_{c_k}} \mathcal{L}\left(\widetilde{\mu}_{c_k}\right) &= \sum_{i,j} \widetilde{z}_{jk} h_{ij} s_{ij} t_i^T - \left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} t_i^T t_i + \lambda_c\right) \widetilde{\mu}_{c_k}, \\ \frac{\partial}{\partial \widetilde{\lambda}_{c_{kl}}} \mathcal{L}(\widetilde{\lambda}_{c_{kl}}) &= \frac{1}{2} \left(\left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} t_{il}^2 + \lambda_c\right) \widetilde{\lambda}_{c_{kl}}^{-2} - \widetilde{\lambda}_{c_{kl}}^{-1}\right). \end{aligned} \tag{B.7}$$

By setting the deviations above to be zeros, we can get

$$\begin{aligned} \widetilde{\mu}_{c_k} &= \left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} t_i^T t_i + \lambda_c I_l\right)^{-1} \left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} s_{ij} t_i^T\right) \\ &= (T \mathbf{\Lambda}(H \widetilde{z}_k) T^T + \lambda_c I_l)^{-1} (T (H \odot S) \widetilde{z}_k), \\ \widetilde{\lambda}_{c_{kl}} &= \sum_{i,j} \widetilde{z}_{jk} h_{ij} t_{il}^2 + \lambda_c, \\ \widetilde{\lambda}_{c_k} &= T \odot T H \widetilde{z}_k + \lambda_c I_l. \end{aligned} \tag{B.8}$$

■

B.3. Proof of Eq. (23)

Proof. We extract all the terms containing $\widetilde{z}_{jk}$, notice that we use the Eq. (A.3) to change the subscript during the extraction, then we have

$$\mathcal{L}(\widetilde{z}_{jk}) = \sum_{i} \widetilde{z}_{jk} \mathcal{L}_1 + \sum_{g}^{k-1} \widetilde{z}_{jk} E_q[\log(1-\theta_g)] + \widetilde{z}_{jk} E_q[\log(\theta_g)] + \sum_{d} \widetilde{z}_{jk} E_q[\log \psi_{kd, x_{jd}}] - \widetilde{z}_{jk} \log \widetilde{z}_{jk}. \tag{B.9}$$

After that, we calculate the deviations of Eq. (B.9), which is

$$\frac{\partial}{\partial \widetilde{z}_{jk}} \mathcal{L}(\widetilde{z}_{jk}) = \sum_{i} \mathcal{L}_1 + \sum_{g}^{k-1} E_q[\log(1-\theta_g)] + E_q[\log(\theta_g)] + \sum_{d} E_q[\log \psi_{kd, x_{jd}}] - \log \widetilde{z}_{jk} - 1. \tag{B.10}$$

By setting the deviation above to zero, we get the updating formula as follows:

$$\widetilde{z}_{jk} \propto \exp\left\{ E_q[\log(\theta_k)] + \sum_{g}^{k-1} E_q[\log(1-\theta_g)] + \sum_{i}^{I} \mathcal{L}_1 + \sum_{d}^{D} E_q[\log \psi_{kd, x_{jd}}] \right\}. \tag{B.11}$$

■

B.4. Proof of Eq. (24)

Proof. We first extract all the terms containing $\widetilde{\psi}_{kd}$:

$$\mathcal{L}(\widetilde{\psi}_{kdm}) = \left(\sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma - \widetilde{\psi}_{kdm}\right) E_q[\log \psi_{kdm}] + \log \mathbf{B}(\widetilde{\psi}_{kd}). \tag{B.12}$$

Then, we calculate the deviation of Eq. B.12:

$$\frac{\partial}{\partial \widetilde{\psi}_{kdm}} \mathcal{L}(\widetilde{\psi}_{kdm}) = \left(\sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma - \widetilde{\psi}_{kdm}\right) \frac{\partial E_q[\log \psi_{kdm}]}{\partial \widetilde{\psi}_{kdm}} - E_q[\log \psi_{kdm}] + \frac{\partial \mathbf{B}(\widetilde{\psi}_{kd})}{\partial \widetilde{\psi}_{kdm}}. \tag{B.13}$$

From Eq. (A.1) and (A.2), it is easy to see that

$$\frac{\partial}{\partial \widetilde{\psi}_{kdm}} \mathcal{L}(\widetilde{\psi}_{kdm}) = \left(\sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma - \widetilde{\psi}_{kdm}\right) \frac{\partial E_q[\log \psi_{kdm}]}{\partial \widetilde{\psi}_{kdm}}. \tag{B.14}$$

Finally, we get the updating formula by setting the deviation above to be zero as follow:

$$\widetilde{\psi}_{kdm} = \sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma. \tag{B.15}$$

■

B.5. Proof of Eq. (25)

Proof. We first extract all the terms containing t_i:

$$\mathcal{L}(t_i) = -\frac{\lambda_t}{2}(t_i - \varphi_i)^T (t_i - \varphi_i) - \sum_{j}^{J} \sum_{k}^{K} \widetilde{z}_{jk} h_{ij} \left(-2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + (t_i^T \widetilde{\mu}_{c_k})^2 + \sum_{l}^{L} \widetilde{\lambda}_{c_k}^{-1} t_{il}^2\right). \tag{B.16}$$

Then, we calculate the deviation of Eq. (B.16) and get

$$\frac{\partial}{\partial t_i} \mathcal{L}(t_i) = -\lambda_t (t_i - \varphi_i) - \sum_{j}^{J} \sum_{k}^{K} \widetilde{z}_{jk} h_{ij} (-s_{ij} \widetilde{\mu}_{c_k} + \widetilde{\mu}_{c_k} \widetilde{\mu}_{c_k}^T t_i + \mathbf{\Lambda}(\widetilde{\lambda}_{c_k}^{-1}) t_i). \tag{B.17}$$

By setting the deviation above to zero, we get the updating formula

$$t_i = \left(\mu_c \mathbf{\Lambda}(\widetilde{Z}^T h_i) \mu_c^T + \mathbf{\Lambda}(\widetilde{\lambda}_c^{-1} \widetilde{Z}^T h_i) + \lambda_t I_l\right)^{-1} \left(\widetilde{\mu}_c \widetilde{Z}^T (h_i \odot s_i) + \lambda_t \varphi_i\right). \tag{B.18}$$

■

Appendix C: The algorithms of generative and optimization process of NDP-JSB.

Algorithm 1 The generative process of the NDP-JSB.

```
for Each job $i$ do
    Draw topic proportion $\varphi_i \sim Dir(\alpha)$
    Draw job latent offset $\epsilon_i \sim N(0, \lambda_t^{-1} I_l)$
    Job latent vector $t_i = \varphi_i + \epsilon_i$
    for Each word $w_{in}$ do
        Draw topic assignment $g_{in} \sim Multi(1; \varphi_i)$
        Draw word $w_{in} \sim Multi(1; \phi_{g_{in}})$
    end for
end for
Draw $\theta_k \sim Beta(1, \beta)$, $k = 1, 2, ..., \infty$.
Group proportion $\pi_k = \theta_k \prod_{b=1}^{k-1} (1 - \theta_b)$, $k = 1, 2, ..., \infty$
Draw company factors for every group $c_k \sim N(0, \lambda_c^{-1} I_l)$, $k = 1, 2, ..., \infty$
Draw company feature distribution parameters for every group $\psi_{kd} \sim Dir(\gamma)$, $k = 1, 2, ..., \infty, d = 1, 2, ..., D$
for Each company $j$ do
    Draw group indicator
    $z_j \sim Multi(1; \pi_1, \pi_2, ..., \pi_\infty)$, $j = 1, 2, ..., J$
    for Each company feature $d$ do
        $x_{jd} \sim Multi(1; \psi_{z_j,d})$, $d = 1, 2, ..., D$
    end for
end for
for Each $(i, j)$ combination do
    Salary $s_{ij} \sim N(t_i^T c_{z_j}, h_{ij}^{-1})$
end for
```

Algorithm 2 The optimization process of the NPD-JSB.

```
Input:
    $W, S, H, X, \alpha, \beta, \gamma, \lambda_c, \lambda_t$
Output: $T, \widetilde{\mu_c}, \widetilde{\lambda}_c, \widetilde{Z}, \varphi, \widetilde{G}, \Phi, \widetilde{\Theta}, \widetilde{\Psi}$
Initialize $T, \widetilde{\mu_c}, \widetilde{\lambda}_c, \widetilde{Z}, \widetilde{\Theta}, \widetilde{\Psi}$ with random values;
Initialize $\varphi, \widetilde{G}, \Phi$ with pre-trained vanilla LDA values to save computation time;
and normalize $\varphi, \widetilde{G}, \Phi, \widetilde{Z}, \widetilde{\Psi}$ to ensure the sum of last dimension equals 1.
while Not Converge do
    Update $\widetilde{\Theta}$ according to Eq. (21)
    Update $\widetilde{\Psi}$ according to Eq. (24) and normalize $\widetilde{\psi}$
    Update $\widetilde{\mu_c}, \widetilde{\lambda}_c$ according to Eq. (22)
    Update $T$ according to Eq. (25)
    Update $Z$ according to Eq. (23) and normalize $Z$
    while NOT Converge do
        Update $\varphi$ according to projection gradient descent method
    end while
    Update $\widetilde{G}$ according to Eq. (6), and normalize the $\widetilde{G}$
    Update $\Phi$ according to Eq. (7), and normalize the $\Phi$
end while
return $T, \widetilde{\mu_c}, \widetilde{\lambda}_c, \widetilde{Z}, \varphi, \widetilde{G}, \Phi, \widetilde{\Theta}, \widetilde{\Psi}$
```

Appendix D: Baselines Introduction

- **SVD**: The idea of SVD in the recommendation scope is to factorize a matrix into two lower-rank latent matrices, where the lower-rank latent matrices can represent the latent information of column- and row-related items, such as movies and viewers. The cross product of two latent matrices can be used for matrix completion. Typically, scientists apply the Frobenius norm regularization on latent matrices from SVD to alleviate the over-fitting problems.

- **NMF**: NMF has similar ideas with SVD in terms of factorizing a matrix into two lower-rank matrices. NMF can also apply to the recommendation system. Different from SVD, NMF bounds values in the latent matrices to be non-negative.

- **PMF**: PMF follows a similar structure, as it also factorizes the matrix into two parts. Different from SVD and NMF, PMF belongs to the probabilistic graphical model. PMF assumes that the values from latent matrices are generated from Gaussian priors, so its inference must align with the probabilistic statistic theories. Changing the priors will change the inference process. The salary prediction module in NDP-JSB follows a PMF structure.

- **CTR**: CTR incorporate text information into the matrix factorization for scientific articles recommendation. The text information from an article is modeled through a latent Dirichlet allocation structure (LDA). CTR combines LDA and PMF under the probabilistic graphic construction, that all variables are generated by some priors that need to be inferred jointly.

- **TADW**: Similar to CTR, TADW incorporates text information into the matrix factorization structure. However, the matrix is factorized into three parts: column-, row- and text-related matrices. In this framework, column- and row-related matrices need to be learned during the optimization process, while the text-related matrix is fixed during the learning. Firstly, the texts are arranged into bag-of-words, then the term frequency–inverse document frequency (tf-idf) transformation (Christian et al. 2016) is used to process them. After that, a separate SVD process is applied to reduce the dimension of texts to obtain the text-related matrix. It is not a probabilistic model, so that every variable does not need to follow a probabilistic distribution.

- **HSBMF**: HSBMF was proposed by (Meng et al. 2018) with the purpose of salary benchmarking on the basis of matrix factorization. A salary matrix will be factorized into two latent matrices to represent the job- and company-related factors, that their cross product can be used for salary prediction. The base structure follows the SVD, where four domain constraints regarding to the job, company, location, and time are utilized to improve the estimation accuracy.

- **BERT-JSB**: BERT (Devlin et al. 2018) is an advanced language learning neural network model, which has been widely used in many natural language processing tasks. As BERT models are computationally expensive in training, some researchers posted pre-trained BERT models for people to use in language processing or customizing neural networks for subsequent tasks. In our experiments, the BERT-JSB embedded with the BERT module for JSB tasks contains four components. First, we selected a pre-trained BERT model[2] to process job descriptions. The output of the BERT module will be 768-dimensional vectors. Second, we use one-hot encoding to process the company-specific information. Then, the output of the BERT module will be concatenated with company-related encodings. Third, the concatenated output will be fed into two fully connected layers with dimensions of 16 and 8 for further processing, where the Relu activation functions are applied. At last, a fully connected prediction layer is used for final salary prediction.

- **Word2Vec-JSB**: Word2Vec is a text embedding model proposed by Le and Mikolov (2014). It is a neural network-based model like BERT, but more efficient in terms of running time. In our experiments, we used the package "gensim"[3] to process the job descriptions, and got 100-dimensional vectors to represent the job description information. Except for the job description learning module, the remaining parts of the Word2Vec-JSB have the same structure and settings as the BERT-JSB model.

[2] `https://github.com/ymcui/Chinese-BERT-wwm`

[3] `https://radimrehurek.com/gensim/models/word2vec.html`

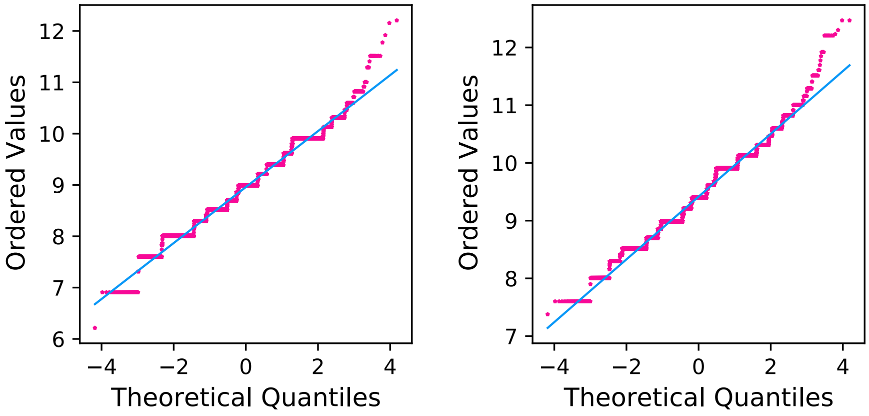
Appendix E: Statistics and Descriptions of Datasets

Table 3 Descriptive statistics of two datasets.

		Salary Statistics			
		Min	Mean	Max	Variance
ItDS	Lower Bound	1,000	9,154	99,000	5,326
	Upper Bound	2,000	15,773	100,000	9,158
FinDS	Lower Bound	500	9,084	200,000	5,771
	Upper Bound	1,000	14,350	260,000	9,206

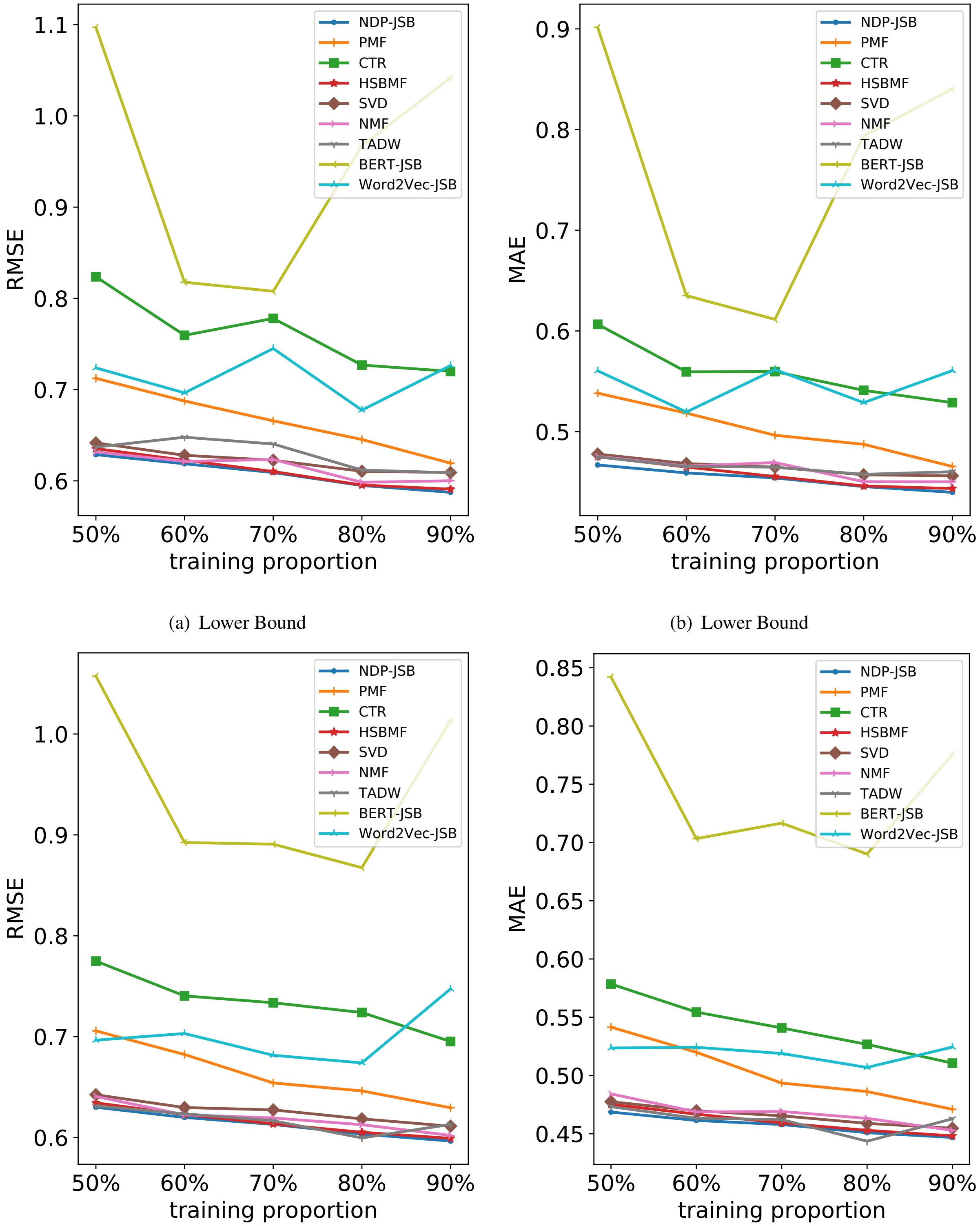
Table 4 The features used in our experiments.

Features	Dimension	Descriptions
Job descriptions (ItDs and FinDs)	5000	Bag-of-words
Work locations (ItDs and FinDs)	5	Shanghai, Beijing, Shenzhen, Guangzhou, Hangzhou
Number of employers (ItDs)	5	$(1,50]$, $(50,150]$, $(150,500]$,$(500,2000]$, $(2000,\infty)$
Number of employers (FinDs)	6	$[1,50)$, $[50,100)$, $[100,500)$, $[500,1000)$, $[1000,10000)$, $[10000,\infty)$
Financial stages (ItDs)	6	No financing or Angel funding, Series-A funding, Series-B funding, Series-C funding, Series-D or plus funding, Public
Industry fields (ItDs)	22	Finance, O2O, Mobile Internet, Data Service, etc.
Owner types (FinDs)	11	Private, Joint Venture, State-owned, Foreign-invested, etc.



(a) FinDS: Lower Bound (b) FinDS: Upper Bound

Figure 15 The probability plots of the logarithmic salaries in FinDS.



(a) Lower Bound

(b) Lower Bound

(c) Upper Bound

(d) Upper Bound

Figure 16 Robust testing results for the different splitting proportions with FinDS.