

## Research Article

# HR Management Big Data Mining Based on Computational Intelligence and Deep Learning

Genliang Zhao <sup>1</sup> and Zhe Xue<sup>2</sup>

<sup>1</sup>*School of International Business and Tourism, Anhui Business College, Wuhu 241002, Anhui, China*

<sup>2</sup>*The Greater Bay Area Huizhou Development Research Institute, Huizhou University, Huizhou 516007, Guangdong, China*

Correspondence should be addressed to Genliang Zhao; [zhaogenliang528@abc.edu.cn](mailto:zhaogenliang528@abc.edu.cn)

Received 17 August 2021; Accepted 28 September 2021; Published 23 October 2021

Academic Editor: Fangqing Wen

Copyright © 2021 Genliang Zhao and Zhe Xue. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Decomposing the structure of a large number of existing posts through data mining will greatly improve the effect of enterprise human resource structure optimization. To this end, this paper proposes an end-to-end competency-aware job requirement generation framework to automate the job requirement generation, and the prediction based on competency themes can realize the skill prediction in job requirements. Then an encoder-decoder LSTM is proposed to implement job requirement generation, and a competency-aware attention mechanism and a replication mechanism are proposed to guide the generation process to ensure that the generated job requirement descriptions comprehensively cover the relevant and representative competency and job skill requirements. A competency-aware strategy gradient training algorithm is then proposed to further enhance the rationality of the generated job requirement descriptions. Finally, extensive experiments on real-world HR data sets clearly validate the effectiveness and interpretability of the proposed framework and its variants compared to state-of-the-art benchmarks.

## 1. Introduction

In order to achieve long-term development, enterprises need to put the advantages of HRs to better use. The modern human capital theory consensus points out that the most valuable asset of an enterprise and the trump card that can win long-term profit in the competitive market and gain maximum benefit for the enterprise is human capital [1]. If an enterprise wants to achieve long-term sustainable development and formulate a more permanent development strategy, it must take precise and detailed HR planning as the first priority, and especially in the current situation of rising human resource costs, only the accurate deployment of human resource costs in advance can reduce costs [2]. Only by making the HR cost planning precise and detailed in advance can, we reduce the cost of manpower and shift to a more efficient cost allocation model [3].

Even if the labor productivity of the best employees in the same position is much higher than that of the average or poor employees, the best employees in the same position should not be taken as the standard for staffing [4]. Only by systematically screening and judging the skills, experience, and level of

different employees as well as the needs of the position can we find the most suitable employees for the position so as to achieve the best overall organizational effectiveness [5, 6]. However, it is worth noting that the process of HR allocation is not a simple selection process but only relies on scientific methods to achieve the best results of the system.

In the use of HR computing intelligence, enterprises are able to access all the contents that are closely related to HR [7]. The relevant data and information on the practical application, on the one hand, and easy to grasp specific information of enterprise development, on the other hand, can provide a reference for the enterprise to make the corresponding management decisions. When data mining technology is applied to HR management, the main content can be divided into three categories: The first category is real-time data. This type of data is mainly reflected in the personnel roster, including individual and organizational levels, where the individual level contains the number of personnel, personnel structure, work experience, age structure, education structure, skills and specialties, certification structure, and family background [8]. The organization level contains

six modules, including HR management, HR strategy management, payroll, and performance management. The second category is dynamic data. This part of data is usually reflected in data reports, such as labor cost tables, and so on [9]. In the management of such data, statistical calculations and tracking records are required. The third category is integrated data. It mainly refers to the information in the form of designed questionnaires and so on and after integration and analysis, such as employee satisfaction.

There are a limited number of personnel at different levels, and too many or too few will affect the stable operation of the company [10, 11]. Therefore, the ratio of supervisors to employees should be kept within a reasonable range. At the same time, in HR management, different management methods are implemented for a given number of employees, and there are differences in the management effect. At the same time, the same management style for different quality and able employees will also make a difference in management efficiency. Therefore, it is crucial for enterprises to adopt scientific and effective management methods according to different information in HR management, while using traditional management methods, it is difficult to realize the use and effective mastering of corresponding information care. In contrast, the management with the assistance of data mining technology in the new era can improve the effect of carrying out the relevant work [12]. For example, if a company controls the proportion of employees who are responsible for the corresponding functions, by analyzing information such as the work capacity of the personnel concerned and the number of people served, it is possible to quickly determine whether the staffing should be increased, maintained, or reduced and improve the rational use of HR [13].

Further, this paper proposes an end-to-end competency-aware neural job requirement generation framework to automate the generation of job requirements, and the prediction based on competency themes enables the prediction of skill words in job requirements. A neural theme model is first designed to explore various competency and skill-related information from real-world HR data. Then an encoder-decoder recurrent neural network is proposed to implement job requirement generation, and a competency-aware attention mechanism and a replication mechanism are proposed to guide the generation process to ensure that the generated job requirement descriptions comprehensively cover the relevant and representative competencies and skill requirements of the job. A competency-aware strategy gradient training algorithm is then proposed to further enhance the rationality of the generated job requirement descriptions. Finally, extensive experiments on real-world HR data sets clearly validate the effectiveness and interpretability of the proposed framework and its variants compared to state-of-the-art benchmarks. Thus, the proposed framework can be effectively applied to talent attraction scenarios in HR services.

## 2. Related Work

Following the commonly used definition, computational intelligence refers to the nontrivial process of identifying novel, potentially useful, and valid patterns in data [9, 14].

During this period, there is a wide range of data mining application areas and corresponding research fields, which also include the field of business management, as well as well-established subfields such as customer management, manufacturing management, or financial management [14].

Recently, these enterprise application domains seem to be complemented by HRM. In the last few years, an increasing number of research contributions aim to support the practical adoption of data mining in HRM. Contributions are various activities and processes of HRM, such as selecting employees or predicting employee turnover [15], to determine the competencies of employees in development, or predicting and evaluating employee performance in performance management [16–18]. To provide these functions, a whole range of data mining methods such as classification trees [19], clustering [20], association analysis [21], support vector machines [22], or neural networks [12, 23] are used, while system improvements and customizations [24] are also presented. In short, browsing the literature gives the impression of a flourishing new field of data mining research that fits the specific requirements of the HR field and is therefore very useful for HR practice.

However, the large number of relevant contributions and different results complicate the overview of the current state of research. Therefore, this thesis aims to design a rational architecture for HRM to be effectively applied to talent attraction scenarios in HR services.

## 3. Data-Mining-Based Multifactor HR Requirements

**3.1. Data Mining.** Data mining is also the effective use of all mathematical algorithms to discover potential patterns from the resulting information. Therefore, it can also be said that the process of finding out the inner laws of the company's HR needs and other influencing elements that interact with each other is to make demand forecasts for the process of data mining in the context of both internal and external effects of the company [25].

Machine learning uses statistics to uncover general patterns that exist in various types of input data and builds training models based on them to predict new input outcomes. For example, support vector machines are based on statistical learning theory, which can reduce structural risk and have the advantages of being theoretically adequate and easy to operate [26].

Initially, support vector machines were proposed in the context of data classification, but the role of kernel functions and support vectors in support vector machines led to the extension of the problem to the field of regression analysis, giving birth to the problem of vector regression machines, also known as support vector regression. The minimum deviation of all sampling points can be obtained in the sample space, and thus, the effect of nonlinear regression in the original space can be derived. The SVR-based features can explore the outstanding performance in the sample data, so it is very useful for enterprises to predict HRs.

The equation defining the regression function is as follows:

$$f(x) = w \times \phi(x) + b. \quad (1)$$

In the high-dimensional eigenspace, the SVR represents the input quantity better by means of the kernel function, while the penalty coefficient  $C$  and the relaxation variable  $\varepsilon$  are introduced together to optimize the daily function as follows:

$$\min \frac{1}{2} \|\omega\|^2 + C \sum (\xi_i + \xi_j^+). \quad (2)$$

The calculation of the extremal point is achieved mainly by means of the Lagrangian function.

**3.2. Variable Weight Support Vector Regression Machines.** When forecasting the demand for HRs, it is necessary to effectively enter the historical data of the time series, which is characterized by a gradual decline with time regression [27, 28]. In the process of regression, the regression error between the earlier data and the new data is almost zero. The weight of the slack variables in the traditional SVR model is the same, and the large variance samples are absolutely dominant in the regression super flat, which allows the regression distortion to appear. With the help of the weight coefficient vector  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ , an identical small penalty strength is taken for all samples, and the importance of early and recent data in the sample series is effectively distinguished so that the regression effects in each sample are scientifically integrated.

The weighting coefficients can be indexed, that is,

$$\lambda_i = ke^{i-N}, \quad (3)$$

where  $N$  is the total number of years of historical data.

## 4. HR Demand Forecasting Case

Taking a company dealing with automobiles as an example, if the HR demand of the company is analyzed, the results generated under this method are tested for prediction. Based on the analysis of correlation, the relevant factors are reasonably selected, and the total output value, total profit, sales situation, and model number are used as the core elements to forecast the HR demand [13, 21].

**4.1. Preprocessing of Data.** If the difference between the numerical magnitudes of the key factors is very obvious, it will cause a serious impact on the serial variance gap of each factor, and if the use is carried out directly, the influences with a large variance will cause a direct impact on the regression results, so it is necessary to preprocess all the data [28]. When processing each group of data, the  $z$ -score method can be used, and its formula is as follows:

$$y_i = \frac{(x_i - \bar{x})}{\delta_\lambda}, \quad (4)$$

where  $x$  is the original data,  $y$  is the predicted value, and  $\delta_\lambda$  is the distribution factor. After preprocessing, the approximate

numerical magnitude of all core factors was effectively obtained.

**4.2. Forecasting Variable-Weight SVR HR Requirements.** The kernel function is filled by a Gaussian function as follows:

$$K(x_i, x_j) = \exp \left[ \frac{-1}{\delta^2 (x_i - x_j)^2} \right]. \quad (5)$$

The experimental findings are carefully analyzed, while the experience accumulated over the years is effectively combined, and the kernel width is set to  $\delta^2 = 2$ , so that the high-dimensional nonlinearity of the data is better represented. When the penalty factor  $C = 100$ , the penalty factor can be avoided, which results in deterioration of performance and generalization of the data. When the base of the relaxation variable in the model is set to 0.01, the accuracy of the data points is very high and the number of support vectors in the training model is minimal, which results in a better extrapolation of the model. In order to enable the prediction accuracy of the method, five years of historical data from 2015 to 2019 were synthesized into a training set [26, 29], which allowed the regression model to be created in a reasonable manner. The actual situation of the company's HRs in 2019 has met the company's strategy implementation needs to the greatest extent, which fully demonstrates the effectiveness of the forecasting method. This is a good indication of the effectiveness of the method. Using this method to forecast the company's HR demand in 2020, the six years of historical data from 2015 to 2020 were combined into a training set, and all key factors for 2020 were entered into the SVR model, resulting in an HR demand of 5,963 people in 2020, with a shortfall of more than 300 people.

## 5. The Proposed Framework

**5.1. Problem Definition.** The goal of this paper is to automate the generation of job requirement descriptions. Given a set  $C$  of job requirement documents for  $|c|$  different jobs, that is,  $c = \{C_i = (X_i, Y_i)\}_{i=1}^{|c|}$ , where  $X_i$  is the job duties, which describe the duties of the  $i$ -th job, and  $Y_i$  is the job requirements, which describe the various competency needs of the job. Specifically, for each job responsibility  $X_i$ , it is assumed to contain  $M^d$  words, that is,  $X_i = \{x_1, x_2, \dots, x_{M^d}\}$  requirements typically contain multiple sentences to describe different competency requirements, so  $Y_i$  each job requirement is represented as  $Y_i = \{y_1, y_2, \dots, y_N\}$ , where  $y_j$  is the  $j$ -th sentence. For example, Figure 1 contains five job requirement statements, that is,  $N=5$ , which correspond to the introduction of education, programming, machine learning, audio processing, and teamwork; the different colors in Figure 1 represent different neurons.

In addition, it is assumed that each  $y_j$  contains  $M_j^c$  words, that is,  $y_j = \{y_{j,1}, y_{j,2}, \dots, y_{j,M_j^c}\}$ . In order to analyze the fine-grained competency requirements of each job, the idea of the paper is followed here to train a neural model

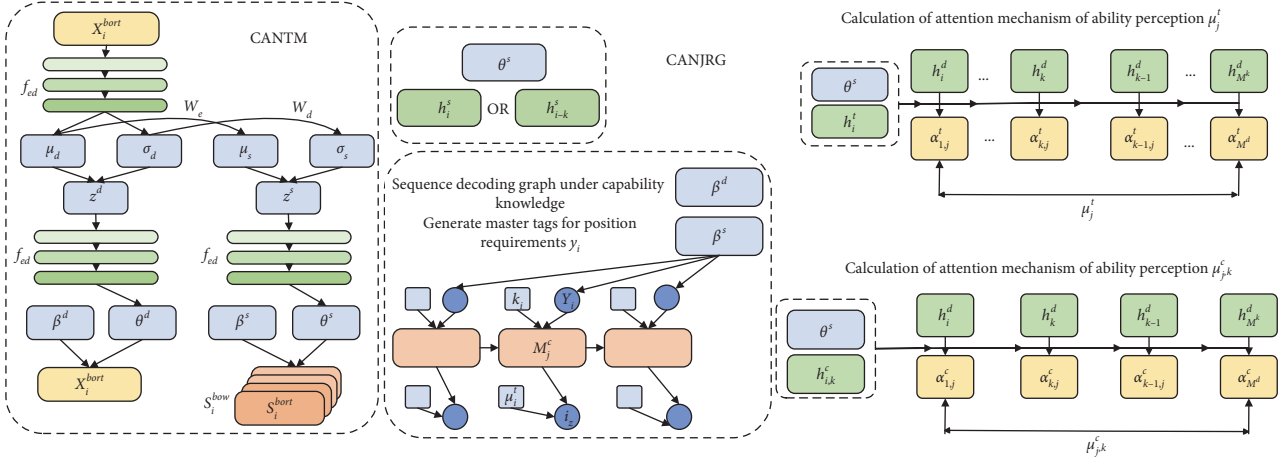


FIGURE 1: Schematic diagram of the framework of this paper (without CAPGTA).

to extract the skill words in each job requirement. Based on the annotation of these words, a list of competency words corresponding to each  $y_j$  can be generated, that is,  $s_j = \{s_{j,1}, s_{j,2}, \dots, s_{j,M_j^s}\}$ . Based on this idea, the following job requirement description generation problem is defined in this section.

**Problem definition:** given a set of HR text blocks  $C$ . Each  $c_i \in C$  contains a job responsibility  $X_i$  and a job requirement  $Y_i$ . The goal of job requirement description generation is to learn a model  $M$  whose smooth and reasonable job requirements  $Y_{new}$  can be generated when a new job responsibility  $X_{new}$  is given.

The proposed automatic job requirement generation framework (Cajon) based on skill prediction contains three main components: capability-aware under neural topic model (CANTM), the neural model for job requirement generation under ability perception (CANJRG), and the policy gradient training algorithm under ability perception (CAPGTA). Figure 1 shows a schematic diagram of the framework without the CAPGTA training algorithm.

**5.2. CANTM.** This subsection proposes a novel CANTM for mining potential competence topics in job responsibilities and job requirements, as shown in Figure 2. Next, the generation process and the inference process in CANTM are described separately. CANTM generation process: in order to model the potential semantics in job responsibilities and job requirements, we assume that there exists two topic spaces with the number of potential topics of  $K^d$  and  $K^s$ . Each topic is divided into  $K_j^d$  and  $K_j^s$ , respectively.

Word distributions  $\beta_j^d$  and  $\beta_j^s$  can be expressed as

$$\begin{aligned} \beta_j^d &= \text{soft max}(\nu_d \cdot (t_d)_j^\top), \\ \beta_j^s &= \text{soft max}(\nu_s \cdot (t_s)_j^\top), \end{aligned} \quad (6)$$

where  $t_d \in R^{K^d \times H}$  and  $t_s \in R^{K^s \times H'}$  are topic-based parameters, respectively, and  $\nu_d \in R^{V^d \times H}$  and  $\nu_s \in R^{V^s \times H'}$  are word-based parameters, respectively, all of which will be learned during the training process. The other  $V^d$  and  $V^s$  are the word list sizes for

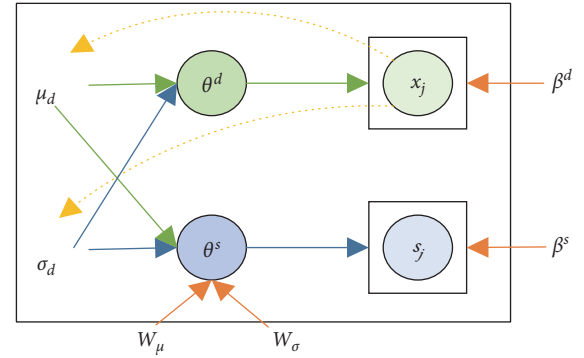


FIGURE 2: Schematic diagrams of CANTM probability diagram (the solid line represents the generation process, and the dashed line represents the variational inference process).

job responsibilities and job requirements, respectively. And only the list of competency words  $s_i$  is considered here as data input for the job requirement part of CANTM, which can reduce input noise and improve the performance of learning potential competency topics in job requirements.

Similar to the LAD topic model [30], it is assumed here that each job duty  $X_i$  and list of competency words  $s_i$  in the job requirements  $Y_i$  have topic vectors  $\theta^d$  and  $\theta^s$ , respectively, where  $\theta^d \in R^{K^d}$  and  $\theta^s \in R^{K^s}$ . Here,  $\theta^d$  and  $\theta^s$  are generated based on Gaussian softmax, respectively. Specifically, the generation process for post  $X_i$  is as follows:

Sampling hidden variables  $z^d \sim \mathcal{N}(\mu_d, \sigma_d^2)$ :

$$\theta^d = \text{soft max}(f_{\theta^d}(z^d)). \quad (7)$$

For the  $L$ -th word in  $X_i$ : sampled word  $x_l \sim \theta^d \cdot \beta^d$ , where  $\mu_d$  and  $\sigma_d^2$  are a priori parameter and  $f_{\theta^d}(\cdot)$  is a neuron activated by a nonlinear function.

The difference is that for the generation process of competency word lists  $S_i$  in job requirements  $Y_i$ , usually, only one competency topic is designed. Based on this, the generation process is as follows:

Sampling hidden variables  $z^s \sim \mathcal{N}(\mu_d, \sigma_d^s)$ :

$$\theta^s = \text{soft max}(f_{\theta^s}(z^s)). \quad (8)$$

The probability of the word  $s_j$  in the  $j$ -th sentence can be expressed as

$$p(s_j | \theta^s, \beta^s) = \theta^s \cdot \prod_{k=1}^{M_j^s} \beta_{*,s_{j,k}}^s, \quad (9)$$

where  $\mu_s$  and  $\sigma_s$  are a priori parameters and  $\beta_{*,s_{j,k}}^s$  represents the column vector of ability words in  $\beta^s$ . In this paper, an end-to-end competency-aware neural job requirement generation framework is proposed to automate the generation of job requirements, and the prediction of skill words in job requirements can be achieved based on the prediction of competency themes. A neural topic model is first designed; then an encoder-decoder LSTM is proposed to implement a job requirement generation, followed by a competency-aware policy gradient-based training algorithm to further enhance the rationality of the generated job requirement descriptions. Finally, extensive experiments on real-world HR data sets clearly validate the effectiveness and interpretability of the proposed framework and its variants in comparison with state-of-the-art benchmarks.

In addition, in order to model the strong correlation between each job position  $X_i$  and the competency term  $S_i$  in the job requirements  $Y_i$ , the following mapping relationship

is assumed here for the a priori parameters of their potential topics.

$\mu_s = W_\mu \mu_d$ ,  $\log \sigma_s = W_\sigma (\log \sigma_d)$  CANTM Inference Process: the edge likelihood [31] of the CANTM-based generation process is as follows:

$$\begin{aligned} p(X_i, S_i | \mu_d, \mu_s, \sigma_d, \sigma_s, \beta^d, \beta^s) \\ = \int p(\theta^d | \mu_d, \sigma_d^2) \int_{k=1}^{M^d} p(x_k | \theta^d, \beta^d) d\theta^d \\ \cdot \int p(\theta^s | \mu_s, \sigma_s^2) \prod_{j=1}^N p(s_j | \theta^s, \beta^s) d\theta^s. \end{aligned} \quad (10)$$

The neural variational method is used here to approximate the posterior distribution on  $\theta^d$  and  $\theta^s$ . Based on equation (10), the variational lower bound for the log-likelihood is as follows:

$$\begin{aligned} \mathcal{L} = \mathbb{E}_{q(\theta^d)} \left[ \sum_{k=1}^M \log p(x_k | \theta^d, \beta^d) \right] - D_{KL}(q(\theta^d) \| p(\theta^d | \mu_d, \sigma_d^2)) \\ + \mathbb{E}_{q(\theta^s)} \left[ \sum_{j=1}^M \log p(s_j | \theta^s, \beta^s) \right] - D_{KL}(q(\theta^s) \| p(\theta^s | \mu_s, \sigma_s^2)), \end{aligned} \quad (11)$$

where  $q(\theta^s)$  and  $q(\theta^d)$  are estimates of the variance distribution of the true posterior  $p(\theta^d | X_i, S_i)$  and  $p(\theta^s | X_i, S_i)$ , respectively.  $D_{KL}$  represents the Kullback-Leibler divergence [5, 32]. The proof is derived as follows:

$$\begin{aligned} \mathcal{L} &= \log \int p(\theta^d | \mu_d, \sigma_d^2) \prod_{k=1}^{M^d} p(x_k | \theta^d, \beta^d) d\theta^d + \log \int p(\theta^s | \mu_s, \sigma_s^2) \prod_{j=1}^N p(s_j | \theta^s, \beta^s) d\theta^s \\ &= \log \int \frac{q(\theta^d)}{q(\theta^d)} p(\theta^d | \mu_d, \sigma_d^2) \prod_{k=1}^{M^d} p(x_k | \theta^d, \beta^d) d\theta^d + \log \int \frac{q(\theta^s)}{q(\theta^s)} p(\theta^s | \mu_s, \sigma_s^2) \prod_{j=1}^N p(s_j | \theta^s, \beta^s) d\theta^s \\ &\geq \int q(\theta^d) \log \left[ p(\theta^d | \mu_d, \sigma_d^2) \prod_{k=1}^{M^d} p(x_k | \theta^d, \beta^d) \right] d\theta^d \\ &\quad + \int q(\theta^s) \log \left[ p(\theta^s | \mu_s, \sigma_s^2) \prod_{j=1}^N p(s_j | \theta^s, \beta^s) \right] d\theta^s \\ &\quad - \int q(\theta^d) \log q(\theta^d) d\theta^d - \int q(\theta^s) \log q(\theta^s) d\theta^s \\ &= \mathbb{E}_{q(\theta^d)} \left[ \sum_{k=1}^{M^d} \log p(x_k | \theta^d, \beta^d) \right] - D_{KL}(q(\theta^d) \| p(\theta^d | \mu_d, \sigma_d^2)) \\ &\quad + \mathbb{E}_{q(\theta^s)} \left[ \sum_{j=1}^N \log p(s_j | \theta^s, \beta^s) \right] - D_{KL}(q(\theta^s) \| p(\theta^s | \mu_s, \sigma_s^2)). \end{aligned} \quad (12)$$



We generate the variance parameters  $\mu_d(X_i, S_i)$ ,  $\log \sigma_d(X_i, S_i)$ ,  $\mu_s(X_i, S_i)$ , and  $\log \sigma_s(X_i, S_i)$  here based on the idea of the paper to estimate  $\mu_d$ ,  $\sigma_d$ ,  $\mu_s$ , and  $\sigma_s$  through input  $X_i$ . This allows the CANTM model to explore potential competency thematic representations through job duties only  $\theta_d$  and  $\theta_s$ . Therefore, an inferential network based on the observed job duties  $X_i$  is introduced here, combined with equation (12) to generate the above variance parameters as follows:

$$\begin{aligned} \mu_d(X_i, S_i) &= f_{\mu_d}(f_{e_d}(X_i^{\text{bow}})), \\ \mu_s(X_i, S_i) &= W_{\mu} \mu_d(X_i, S_i), \\ \log \sigma_d(X_i, S_i) &= f_{\sigma_d}(f_{e_d}(X_i^{\text{bow}})), \\ \log \sigma_s(X_i, S_i) &= W_{\sigma} \log \sigma_d(X_i, S_i), \end{aligned} \quad (13)$$

where  $(X_i^{\text{bow}})$  is the bag-of-words vector of  $X_i$ ,  $(f_{e_d}(\cdot))$  is a neuron activated by a nonlinear function, and  $(f_{\mu_d}(\cdot))$  and  $(f_{\sigma_d}(\cdot))$  are linear neural perceptual functions.

Based on this, the following loss function can be directly minimized for each set of instances  $(X_i, S_i)$  during the training process:

$$\mathcal{L}_{\text{CANTM}} = - \sum_{k=1}^{M^d} \log(\theta^d \cdot \beta_{*, x_k}^d) + D_{\text{KL}}(q(\theta^d) \| p(\theta^d)). \quad (14)$$

Therefore, all parameters in CANTM can be inferred, and the potential competency themes involved in each position can be further explored.

**5.3. CANJRG.** After learning about the potential competency themes through CANTM, this subsection describes how to use the encoder-decoder neural model to generate job requirements. As shown in Figure 3, it contains two main components, including a sequence encoder that extracts semantic information from the input job responsibilities  $X_i$ , and a sequence decoder under competency awareness, which can generate each word in the job requirements  $Y_i$  by the guidance of potential competency themes.

**Sequence encoder:** first use an embedding layer to find the embedding vector  $e_k^d$  for each word  $x_k$  in  $X_i$  and then use a Bi-LSTM [5, 33] to encode the sequence  $\{e_1^d, e_2^d, \dots, e_{M^d}^d\}$ :  $\overrightarrow{h_k^d} = \text{LSTM}(e_k^d, \overrightarrow{h_{k-1}^d})$ ,  $\overleftarrow{h_k^d} = \text{LSTM}(e_k^d, \overleftarrow{h_{k+1}^d})$ , where  $e_k^d = W_{e^d} x_k$  is the word vector of  $x_k$  and the LSTM is a unidirectional LSTM network. Finally,  $h_k^d = [\overrightarrow{h_k^d}; \overleftarrow{h_k^d}]$  is used to represent the final hidden vector of the  $x_k$  sequence encoder.

**Competency-aware sequence decoder:** the following describes how to construct a decoder to generate each word in job requirement  $Y_i$ . In the generation process, the competency topic  $t_j$  is first estimated for each sentence  $y_j$  in  $Y_i$ , and then each word is predicted the following probability  $y_{j,k}$ :

$$p(y_{j,k} | X_i) = p(y_{j,k} | y_{<j}, y_{j,<k}, \mathcal{H}, \theta^s, t_j), \quad (15)$$

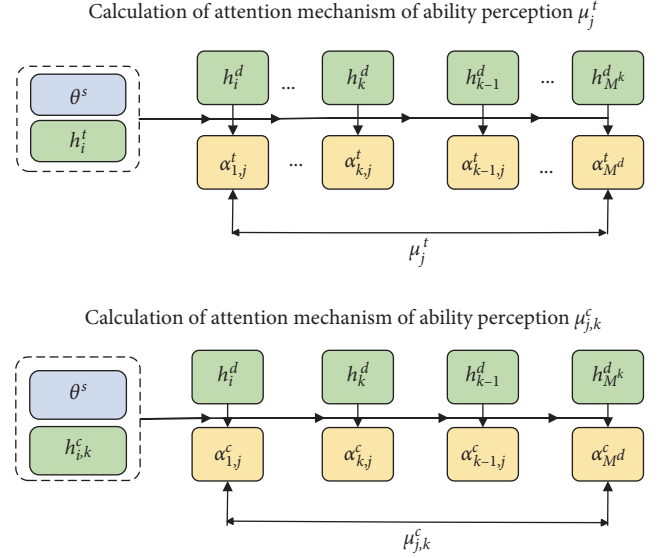


FIGURE 3: Schematic diagrams of the attention mechanism under ability perception: (a) calculation of attention mechanism of ability perception  $\mu_j^t$  and (b) calculation of attention mechanism of ability perception  $\mu_{j,k}^c$ .

where  $y < j$  represents the sequence  $\{y_1, y_2, \dots, y_{j-1}\}$ ,  $y_{j,<k}$  and represents  $\{y_{j,1}, y_{j,2}, \dots, y_{j,k-1}\}$  and  $\mathcal{H} = \{h_1^d, h_2^d, \dots, h_{M^d}^d\}$  represents the implicit state of all sequence encoders.  $\theta^s$  is  $Y_i$ ; the implicit capability topic vector learned through CANTM, and  $t_j \in [1, K^s]$  is the topic label for each utterance  $y_j$ .

Specifically, the sequence decoder under capability awareness is constructed based on two one-way LSTMs [34].  $h_j^t$  and  $h_{j,k}^c$  represent the implicit status of the competent topics  $t_j$  and words  $y_{j,k}$  computed by the LSTM, respectively.

$$\begin{aligned} h_j^t &= \text{LSTM}\left(\left[e_{j-1}^t; \theta^s; h_{j-1, M_{j-1}^c}^c\right], h_{j-1}^t\right) h_{j,k}^c \\ &= \text{LSTM}\left(\left[e_{j,k-1}^c; \theta^s; h_j^t\right], h_{j,k-1}^c\right), \\ e_j^t &= W_{e^t} t_j, e_{j,k}^c = W_{e^c} y_{j,k}, \end{aligned} \quad (16)$$

where  $e_j^t$  and  $e_{j,k}^c$  is the embedded representations of  $t_j$  and  $y_{j,k}$ .  $M_j^c$  is the length of statement  $y_j$ . Also, implicit states  $h_0^t$  and  $h_{1,0}^c$  are initialized by  $h_{M^d}^d$ , and  $h_{j,0}^c$  is initialized by  $h_{j-1, M_{j-1}^c}^c$ . In addition, two capability-aware attention mechanisms are designed here to capture contextual features from  $H$  to enhance the performance of the generation process as follows:

$$\begin{aligned} \alpha_{i,j}^t &= \frac{\exp(g_{i,j}^t)}{\sum_{p=1}^N \exp(g_{p,j}^t)}, \\ \alpha_{i,j,k}^c &= \frac{\exp(g_{i,j,k}^c)}{\sum_{p=1}^{M_j^c} \exp(g_{p,j,k}^c)}, \end{aligned} \quad (17)$$

$$g_{i,j}^t = v_{\alpha^t}^\top \tanh(W_{\alpha^t} [h_i^d; h_j^t; \theta^s] + b_{\alpha^t}),$$

$$g_{i,j,k}^c = v_{\alpha^c}^\top \tanh(W_{\alpha^c} [h_i^d; h_{j,k}^c; \theta^s] + b_{\alpha^c}).$$

The capability-aware context vectors  $u_j^t$  and  $u_{j,k}^c$  can then be calculated by the following equations:

$$\begin{aligned} u_j^t &= \sum_{l=1}^N \alpha_{l,j}^t h_l^d, \\ u_{j,k}^c &= \sum_{l=1}^{M_j^c} \alpha_{l,j,k}^c h_l^d. \end{aligned} \quad (18)$$

The ability theme labels can then be predicted.  $t_j$  and  $y_j$  for each word in  $y_{j,k}$  as follows:

$$\begin{aligned} p(t_j | t_{<j}, \mathcal{H}, \theta^s) &= \text{soft max}(W_t[h_j^t; u_j^t; \theta^s] + b_t), \\ p(y_{j,k} | y_{<j}, y_{j,<k}, \mathcal{H}, \theta^s, t_j) &= \text{soft max}(W_c[h_{j,k}^c; u_{j,k}^c; \theta^s] + b_c). \end{aligned} \quad (19)$$

In addition, an ability-aware replication mechanism is designed such that the proposed decoder can directly replicate the words in the ability vocabulary. Specifically, a generation probability is defined here when  $y_j$  generates the  $k$ -th word:  $y_{j,k}; p_{\text{gen}} \in [0, 1]$ .

The probability distribution of the predicted words based on the ability word list can then be updated with the following equation:

$$\begin{aligned} p(y_{j,k} | y_{<j}, y_{j,<k}, \mathcal{H}, \theta^s, t_j) \\ = p_{\text{gen}} \text{soft max}(W_c[h_{j,k}^c; u_{j,k}^c; \theta^s] + b_c) + (1 - p_{\text{gen}})(\beta^s)_{t_j}, \end{aligned} \quad (20)$$

where  $(\beta^s)_{t_j}$  is the word distribution in topic.

Finally, in the heterogeneous model, for each group of training instances  $(X_i, Y_i)$ , the parameters in the model are learned by minimizing the following cross-entropy loss function:

$$\begin{aligned} \mathcal{L}_{\text{CTL}} &= - \sum_{j=1}^N \log p(t_j | X_i, \theta_s), \\ \mathcal{L}_{\text{GJR}} &= - \sum_{j=1}^N \sum_{k=1}^{M_j^s} \log p(y_{j,k} | X_i, \theta_s). \end{aligned} \quad (21)$$

**5.4. Capability-Aware Policy Gradient Training Algorithm (CAPGTA).** Before introducing CAPGTA, a basic end-to-end training approach will be shown to learn all the parameters in the above two models. Specifically, because of the CANTM neural variation, the loss functions  $\mathcal{L}_{\text{CANTM}}$ ,  $\mathcal{L}_{\text{CTL}}$  and  $\mathcal{L}_{\text{GJR}}$ , can be trained jointly at the same time.

$\mathcal{L}^* = \mathcal{L}_{\text{GJR}} + \lambda_1 \mathcal{L}_{\text{CANTM}} + \lambda_2 \mathcal{L}_{\text{CTL}}$ , where  $\lambda_1$  and  $\lambda_2$  are hypermastigote to balance each model. The teacher forcing algorithm is used in the training process, that is, the previous

real word  $y_{j,k-1}$  is used in the training to calculate  $h_j^t$  and  $h_{j,k}^c$ . For the ability topic  $t_{j-1}$ , the following is used for generation:

$$t_{j-1} = \arg \max_l \left\{ (\theta^s)_l \left( \prod_{k=1}^{M_{j-1}^s} \beta_{*, s_{j-1,k}}^s \right) \right\}_{l=1}^{K^s}. \quad (22)$$

And the predicted values are used as input in the testing sessions.

Direct minimization  $\mathcal{L}^*$  does not always generate the best job requirements because it does not directly optimize discrete assessment metrics such as ROUGE and BLKJ [35]. In addition, it is desired here that the accuracy of the competencies involved in the generated job requirements can be optimized more intuitively, so that the rationality and validity of the generated results can be better ensured.

Some recent reinforcement learning techniques can be used to solve this nondifferentiable task metric problem. Here, the combination of CANTM and CANJRG can be considered as an agent [30, 36], which interacts with the environment, that is, the training instances. Given an input job duty  $X$ , strategy  $(\text{pol} - \text{icy})p_\theta(y_{j,k}X, \hat{y}_{<j}, y_{j,<k})$  is determined by the parameter  $\theta$  of the intelligence for each action, that is, predicting the next word based on the current state. Until the end position (EOS) of the sequence of job requirements is generated, a reward will be observed. The goal of the whole training is to learn the strategy by minimizing the negative expected reward of

$$\mathcal{L}^{\text{RL}} = -\mathbb{E}_{\hat{Y} \sim p_\theta(\hat{Y}|X)} [r(\hat{Y})]. \quad (23)$$

Based on reinforcement learning algorithm, it obtains that

$$\nabla_\theta \mathcal{L}^{\text{RL}} = -\mathbb{E}_{\hat{Y} \sim p_\theta(\hat{Y}|X)} [r(\hat{Y}) \nabla_\theta \log p_\theta(\hat{Y} | X)]. \quad (24)$$

It is possible to use a simple Monte-Carlo sampling  $\hat{Y}$  based strategy  $p_\theta$  as follows:

$$\begin{aligned} \nabla_\theta \mathcal{L}^{\text{RL}} &\approx -r(\hat{Y}) \nabla_\theta \log p_\theta(\hat{Y} | X) \\ &= -r(\hat{Y}) \nabla_\theta \sum_{j=1}^N \sum_{k=1}^{M_j^c} \log p_\theta(y_{j,k}X, \hat{y}_{<j}, y_{j,<k}) \\ &= -r(\hat{Y}) \nabla_\theta \sum_{j=1}^N (\log p_\theta(\hat{t}_j | X, \hat{t}_{<j}) \\ &\quad + \sum_{k=1}^{M_j^c} \log p_\theta((y_{j,k}X, \hat{y}_{<j}, y_{j,<k}))) \end{aligned} \quad (25)$$

where  $(\hat{t}_1, \dots, \hat{t}_N)$  is a Monte-Carlo [37] sample of the capability label.  $p_\theta(\hat{t}_j | X, \hat{t}_{<j})$  and  $p_\theta((y_{j,k}X, \hat{y}_{<j}, y_{j,<k}))$  are calculated from equations (23) and (24), respectively.

$$\nabla_\theta \mathcal{L}^{\text{RL}} \approx (r(\hat{Y}^g) - r(\hat{Y})) \nabla_\theta \sum_{j=1}^N \left( \log p_\theta(\hat{t}_j | X, \hat{t}_{<j}) + \sum_{k=1}^{M_j^c} \log p_\theta(y_{j,k}X, \hat{y}_{<j}, y_{j,<k}) \right). \quad (26)$$

As mentioned earlier, here, it is desired to directly optimize the accuracy of the competencies in the generated job requirements. Therefore, we use the F1 values [38] of the generated skill terms as a reward function, that is,

$$r_{\text{capability}}(\hat{Y}) = \frac{2\text{Card}(\hat{S} \cap S)}{\text{Card}(\hat{S}) + \text{Card}(S)}, \quad (27)$$

where  $S$  is the set of skill words in the actual job requirements and  $\hat{S}$  is the set of skill words in  $\hat{Y}$ , representing the set size. The ROUGE-1 score is also incorporated into the reward function, which is used to measure statistical information based on the longest common subsequence between the actual and model-generated job requirements. This allows a direct optimization of the similarity of the sentence hierarchy to authenticity, which helps improve the fluency of the generated text. The reward function can then be set to

$$r(\hat{Y}) = \lambda_3 r_{\text{capability}}(\hat{Y}) + \text{Rouge} - 1(\hat{Y}, Y). \quad (28)$$

Finally,  $\mathcal{L}^*$  and  $\mathcal{L}^{\text{RL}}$  are used jointly to obtain the overall learning objective function as follows:

$$\mathcal{L} = (1 - \gamma)\mathcal{L}^* + \gamma\mathcal{L}^{\text{RL}}, \quad (29)$$

where  $\gamma$  is the dynamic hyperparameter during the training process. It is first set to 0 for a period of training alone  $\mathcal{L}^*$ , and then the value is gradually increased.

## 6. Experimental Analysis

This section presents the results of extensive experiments of quantitative analysis and manual evaluation on real-world HR data sets [4, 12] to demonstrate the effectiveness of the proposed Cajon in job skill prediction and job requirement generation.

**6.1. Experimental Data.** Two real-world HR data sets [4, 12], including technical ( $T$ ) and product ( $P$ ) related job data sets, are used here. Specifically, 3,475 and 2,351 different jobs were collected, respectively, including their job responsibilities and corresponding job requirement texts, which have been carefully proofread by six HR experts to ensure fluency and reasonableness. Some statistics are shown in Table 1 and Figures 4 and 5. In the experiments, 80% of the data set were randomly selected as training data, another 10% as test data to verify the performance, and the last 10% was used to tune the parameters.

In addition to generating reasonable skill words in job requirements, LSTM-CRF [15, 25] model was trained to extract possible competency words based on the method of the paper. With the help of HR experts, a final vocabulary containing 4,825 skill entities was obtained.

**6.2. Training Parameters and Environment Setting.** In the competency-aware neural topic model, the raw input of job responsibilities and competency words from the job descriptions is first converted into a bag-of-words vector [4, 23]. And before that, deactivated words and high- and low-frequency words are removed to enhance the model. The performance of

the model is enhanced by removing the discontinued words and high- and low-frequency words. Here, the number of topics ( $K^d, K^s$ ) is set to (30, 50) and (30, 30) for the  $T$  and  $P$  data sets, respectively. In addition, we add batch normalization in computing  $\mu_d(X_i, S_i), \mu_s(X_i, S_i), \log \sigma_d(X_i, S_i)$ , and  $\log \sigma_s(X_i, S_i)$  to avoid the problem that KL divergence disappears during training.

In the capability-aware postrequirement generation model, the embedding layer sizes of word  $x_k, y_{j,k}$ , and topic tag  $t_k$  are 128, 128, and 50, respectively. Sequence encoder is implemented by a bidirectional LSTM, and the hidden layer size of each LSTM layer is 256. Capability-aware sequence decoder is implemented by two unidirectional LSTMs, both of which have a hidden layer size of 256. In addition, the size of the hidden layer states in both the capability-aware attention mechanism and the capability-aware replication mechanism are also set to 256.

During the training of the complete Cajon framework, the parameters are initialized using the Xavier strategy.

Then 200 rounds of pretraining are performed on CANTM. After that, we set  $\lambda_1 = 1$  and  $\lambda_2 = 1$  to train the part of Cajon other than the reinforcement learning loss function. Finally, we set  $\lambda_3 = 1$  and incrementally increase to train our model by equation (12). In addition, Adam is used as the optimizer, and the initial learning rate is set to 0.001. And, the gradient crop is also set to 1.0 to stabilize the training process. In the test phase of generation, we used the Beam Search algorithm and set the cluster size to 4.

The overall experiments were performed on a Linux server configured with RedHat 4.8.536, 2.40 GHz Intel(R) Xeon(R) Gold6148 CPU; models were developed based on the tensor flow framework.

**6.3. Benchmarking Algorithm.** To evaluate the effectiveness of the proposed approach, several state-of-the-art text generation methods are compared here, and these methods are adapted to fit the problem definition setting.

Seq2Seq [14] is a classical text-to-text generation model, which was proposed in the paper with the aim of implementing neural machine translation. In the experiments of this section, a concat-based approach to compute attention mechanisms is also applied, which is similar to the approach proposed in this chapter.

Kit [18] is a variant of Seq2Seq, a model that implements a pointer network and an overlay mechanism to handle the automatic digest problem.

Kid is a natural language generation model based on transformer networks, which are proposed to solve the sequence-to-sequence generation problem.

In addition, state-of-the-art automated job description writing methods are compared.

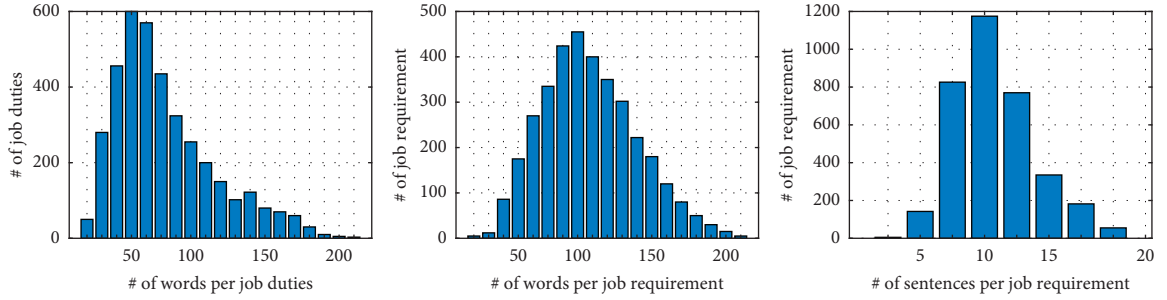
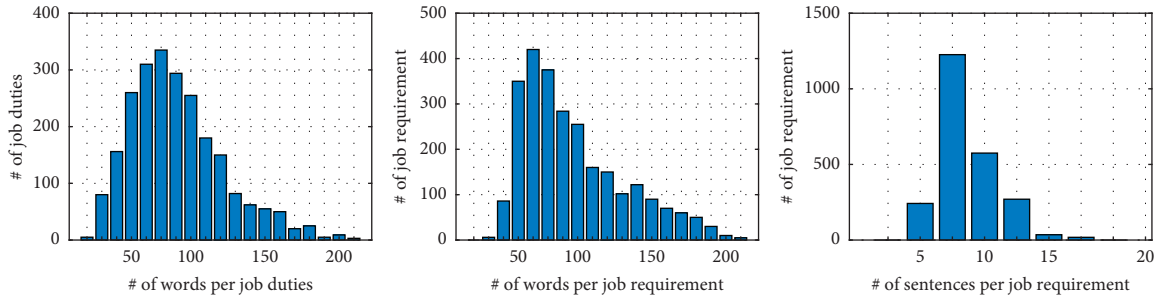
SAMA [19, 21] is the state-of-the-art automated job description writing model, which is presented in the paper. For a fair comparison with the proposed model, the characteristics of the additional information it uses (e.g., company size) are removed in this section of the experiments.

In addition, four variants of the Cajon framework are compared to assess the impact of each component of the



TABLE 1: Statistical description of  $T$  and  $P$  data sets.

Data set	Statistical indicators	Value
$T$	Total number	3,475
	Average number of words for job duties	50.11
	Average word count of job requirements	100.10
	Average number of words of ability for job requirements	25.72
	Average number of sentences for job requirements	6.95
$P$	Total number	2,351
	Average number of words for job duties	74.98
	Average word count of job requirements	75.80
	Average number of words of ability for job requirements	15.58
	Average number of sentences for job requirements	4.98

FIGURE 4: Text statistics of the technology ( $T$ ) data set.FIGURE 5: Text statistics of the product ( $P$ ) data set.

model on the generated results: Cajon (w/o RL) is a variant of Cajon in which the CAPGTA is removed from the training, that is, the training is done directly by the formula. Cajon (w/o RL,  $\mathcal{L}_{\text{CANTM}}$ ) is a variant of Cajon (w/o RL,  $\mathcal{L}_{\text{CTL}}$ ) that removes the ability topic label related part of the sequence decoder, that is, only  $\theta^s$  is used to introduce ability topic information.

Cajon (w/o RL, topic-copy) is a variant of Cajon (w/o RL), which removes the capability-aware replication-based mechanism.

**6.4. Evaluation Indicators.** In order to evaluate the effectiveness of job requirement generation, both automatic and manual assessments were used.

In the automatic evaluation, standard ROUGE metrics were used, including ROUGE-1, ROUGE-2, and ROUGE-L, which measure the statistics of unary word overlap, binary word overlap, and longest common subsequence (LCS) [31]

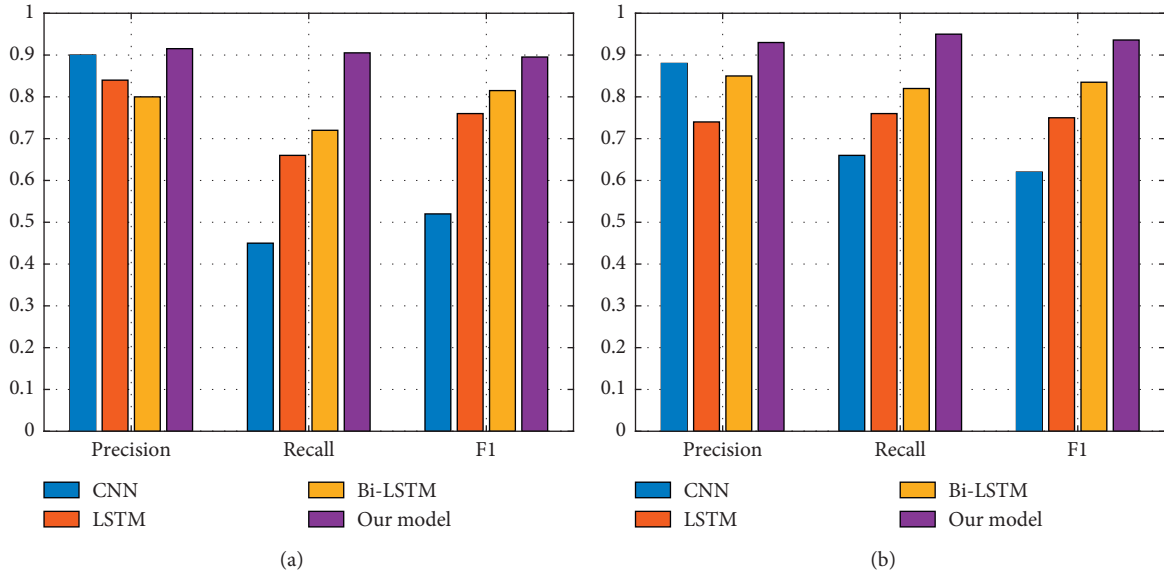
in the comparison of real and automatically generated results, respectively. The BLEU evaluation metric, which measures the cooccurrence of  $n$ -words, was also used. Finally, the precision rate, recall rate, and F1 value of skill words in job requirements are used to automatically verify the rationality and validity of the generated results, as shown in Table 2.

Figure 6 shows the accuracy, recall, and F1 value data set of Cajon and its variants; the proposed model improves 1.06% and 4.60% in automatic metrics ROUGE-1 and BLEU-1, and 3.00% and 7.16% in manual metrics Fluency and Validity, respectively, compared to the best available techniques. This result clearly demonstrates the effectiveness of the proposed model in generating fluent and reasonable job requirements [39].

In addition, Figure 6 shows the precision, recall, and F1 values of the generated competency words in the job requirements. Here, it can be found that the proposed model outperforms the best results of all benchmarks of 9.49%,

TABLE 2: Experimental results of Cajon and comparison methods.

Data set	Method	ROUGE			BLEU		Manual evaluation metrics	
		ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1 (%)	BLEU-4 (%)	Fluency	Validity
<i>T</i>	Seq2Seq	0.6389	0.4151	0.5996	37.32	16.22	3.66	3.58
	pGNet	0.6413	0.4246	0.6061	38.14	16.75	3.70	3.61
	UniLM	0.6167	0.3965	0.5863	34.64	12.29	3.52	3.14
	SAMA	0.6004	0.3277	0.5455	39.30	15.07	3.47	3.52
	Cajon	0.6531	0.4296	0.6077	41.81	19.41	3.73	3.81
<i>P</i>	Seq2Seq	0.7236	0.5085	0.7138	32.02	7.28	3.63	3.34
	pGNet	0.7421	0.5410	0.7331	32.72	8.28	3.67	3.41
	UniLM	0.7542	0.5439	0.7452	31.43	6.42	3.46	3.28
	SAMA	0.7160	0.4788	0.7074	34.76	9.00	3.55	3.49
	Cajon	0.7522	0.5757	0.7531	36.46	10.84	3.78	3.74

FIGURE 6: Different model performance: (a) technology (*T*) data set and (b) product (*P*) data set.

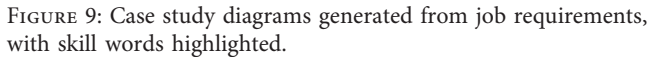
3.55%, and 6.73% in the technical data and 20.62%, 5.29%, and 17.69% in the product data set, respectively. It clearly validates that the generated results of the proposed framework can more accurately capture the relevant and representative skill requirements of the position.

**Ablation experiments:** here, the effects of the proposed model and its variants are compared. And Seq2Seq can also be used as a variant of the proposed method, that is, the CANTM model is removed. Obviously, it is clear from the results that all model components can enhance the effect. Specifically, the performance decreases rapidly when only potential capability topic information is considered, which proves the importance of predicting potential capability topic labels for the decoder. As shown in Figure 7, the capability-awareness-based attention mechanism can improve about 2.61% and 1.38% of ROUGE-1 and BLEU-1, respectively, in the technology data set, and 2.53% and 4.83% in the product data set, respectively. Meanwhile, the capability-awareness-based replication mechanism can improve 1.87% and 0.84% in the technical data set on ROUGE-1 and BLEU-1 while spoon effect and 2.92% and 1.54% in the product data set,

respectively. In addition, Figure 8 shows that the proposed CAPGTA can effectively improve the precision, recall, and F1 values of skill words in generating job requirements.

**Subject number parameter experiments:** as shown in Figure 8, to evaluate the parameter sensitivity, Cajon is trained here by tuning parameters  $K^d$  and  $K^s$ , 0 to 100, whose other ones are fixed at  $K^d = 30$  and  $K^s = 50$  in the technical data set and  $K^d = 30$  and  $K^s = 30$  in the product data set. Here, it can be clearly observed that the best results can be obtained in the technical data set  $K^d = 30$  and  $K^s = 30$  and in the product data set  $K^s = 30$  and  $K^d = 30$ .

**6.5. Generating Example Studies and Discussion.** To further illustrate the effectiveness and interpretability of the proposed framework, an example of job requirements generated by Cajon is given in Figure 9. Given the position to hire a data mining algorithm engineer, it can be found that the generated results are fluent and include competency requirements regarding education, work experience, data mining algorithms, basic programming language, and



effectively guide the generation of job requirement texts, thus demonstrating the interpretability of the proposed framework.

In this paper, an end-to-end competency-aware neural job requirement generation framework is proposed to automate the generation of job requirements, and the prediction of skill words in job requirements can be achieved based on the prediction of competency themes. Then an encoder-decoder recurrent neural network is proposed to implement a job requirement generation, followed by a competency-aware policy gradient-based training algorithm to further enhance the rationality of the generated job requirement descriptions. Finally, extensive experiments on real-world HR data sets clearly validate the effectiveness and interpretability of the proposed framework and its variants in comparison with state-of-the-art benchmarks.

The data sets used in this paper are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

## Acknowledgments

This work was supported by the Outstanding Young Scholars Program (2020): "Research on the Driving Mechanism, Model Selection and Path Optimization of the Transformation and Upgrading of the Human Resource Service Industry in Anhui Province in the AI Era" under grant no. gxyq2020229.

## References

- [1] R. Iqbal, F. Doctor, B. More, S. Mahmud, and U. Yousuf, "Big data analytics and computational intelligence for cyber-physical systems: recent trends and state of the art applications," *Future Generation Computer Systems*, vol. 105, pp. 766–778, 2020.
- [2] R. Iqbal, F. Doctor, B. More, S. Mahmud, and U. Yousuf, "Big data analytics: computational intelligence techniques and application areas," *Technological Forecasting and Social Change*, vol. 153, p. 119253, 2020.
- [3] Y. E. Zhang, S. Liu, and S. Xu, "Integrating the split/analyze/meta-analyze (SAM) approach and a multilevel framework to advance big data research in psychology," *Zeitschrift für Psychologie*, vol. 226, 2019.
- [4] R. Li, Q. Liu, and J. Gui, "Indoor relocalization in challenging environments with dual-stream convolutional neural networks," *IEEE Transactions on Automation Science and Engineering*, vol. 15, pp. 651–662, 2017.
- [5] N. Chen, W. Liu, R. Bai, and A. Chen, "Application of computational intelligence technologies in emergency management: a literature review," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 2131–2168, 2019.
- [6] A. Chadha, A. Abbas, and Y. Andreopoulos, "Video classification with CNNs: using the codec as a spatio-temporal activity sensor," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 475–485, 2017.
- [7] R. Arandjelovic, P. Gronat, and A. Torii, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307, Las Vegas, NV, USA, June 2016.
- [8] F. L. Oswald, T. S. Behrend, D. J. Putka, and E. Sinar, "Big data in industrial-organizational psychology and human resource management: forward progress for organizational research and practice," *Annual Review of Organizational Psychology and Organizational Behavior*, vol. 7, no. 1, pp. 505–533, 2020.
- [9] N. Nawaz, "Exploring artificial intelligence applications in HR management," *Journal of Management Information and Decision Sciences*, vol. 23, no. 5, pp. 552–563, 2020.
- [10] I. Harris and M. B. Küssner, "Come on baby, light my fire: sparking further research in socio-affective mechanisms of music using computational advancements," *Frontiers in Psychology*, vol. 11, p. 3274, 2020.
- [11] G. Nagarajan and R. I. Minu, "Multimodal fuzzy ontology creation and knowledge information retrieval," in *Proceedings of the International Conference on Soft Computing Systems*, pp. 697–706, Delhi, India, November 2016.
- [12] J. Ock and H. An, "Machine learning approach to personality assessment and its application to personnel selection," *Korean Journal of Industrial and Organizational Psychology*, vol. 34, no. 2, pp. 213–236, 2021.
- [13] Z. Gao, P. Wang, H. Wang, M. Xu, and W. Li, "A review of dynamic maps for 3D human motion recognition using ConvNets and its improvement," *Neural Processing Letters*, vol. 52, no. 2, pp. 1501–1515, 2020.
- [14] A. Agus, S. K. Krishnan, and S. L. S. A. Kadir, "The structural impact of total quality management on financial performance relative to competitors through customer satisfaction: a study of Malaysian manufacturing companies," *Total Quality Management*, vol. 11, no. 4–6, pp. 808–819, 2000.
- [15] Y. Zhao, M. K. Hryniewicki, F. Cheng, B. Fu, and X. Zhu, "Employee turnover prediction with machine learning: a reliable approach," in *Proceedings of the SAI Intelligent Systems Conference*, pp. 737–758, London, UK, September 2018.
- [16] L. Wang, C. Zhang, and Q. Chen, "A communication strategy of proactive nodes based on loop theorem in wireless sensor networks," in *Proceedings of the 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 160–167, Wanzhou, China, November 2018.
- [17] H. He, C. Maple, and T. Watson, "The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1015–1021, Vancouver, Canada, July 2016.
- [18] S. N. Mishra and D. R. Lama, "A decision making model for HR management in organizations using data mining and predictive analytics," *International Journal of Computer Science and Information Security*, vol. 14, no. 5, p. 217, 2016.
- [19] N. Hajiheydari, S. H. Khabiri, and M. Talafi Daryani, "A framework for data mining approach applications in HR management," *Iranian Journal of Management Sciences*, vol. 12, no. 47, pp. 21–50, 2017.
- [20] Y. Zhang, S. Xu, L. Zhang, and M. Yang, "Big data and human resource management research: an integrative review and new directions for future research," *Journal of Business Research*, vol. 133, pp. 34–50, 2021.
- [21] R. Etukudo, *Strategies for Using Analytics to Improve HR Management*, Walden University, Minneapolis, Minnesota, 2019.
- [22] X. Wang, S. Wang, Z. Huang, and Y. Du, "Condensing the solution of support vector machines via radius-margin bound," *Applied Soft Computing*, vol. 101, p. 107071, 2021.
- [23] M. Ameer, S. P. Rahul, and S. Manne, "HR analytics using power Bi visualization tool," in *Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1184–1189, Madurai, India, May 2020.
- [24] J. Okolita, *Moving from Instincts to Insights: Leaders Behavioral Intention to Use People Analytics*, The Chicago School of Professional Psychology, Chicago, IL, USA, 2019.
- [25] D. J. M. Reddy, S. Regella, and S. R. Seelam, "Recruitment prediction using machine learning," in *Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–4, Bihar, India, October 2020.
- [26] N. A. Rahmad, N. A. J. Sufri, N. H. Muzamil, and M. A. As'ari, "Badminton player detection using faster region convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1330–1335, 2019.



- [27] C. Zhang, T. Xie, K. Yang et al., "Positioning optimisation based on particle quality prediction in wireless sensor networks," *IET Networks*, vol. 8, no. 2, pp. 107–113, 2019.
- [28] S.-C. Kang and S. A. Snell, "Intellectual capital architectures and ambidextrous learning: a framework for human resource management," *Journal of Management Studies*, vol. 46, no. 1, pp. 65–92, 2009.
- [29] P. Junni, R. M. Sarala, V. Taras, and S. Y. Tarba, "Organizational ambidexterity and performance: a meta-analysis," *Academy of Management Perspectives*, vol. 27, no. 4, pp. 299–312, 2013.
- [30] H. Yu, Q. Wang, and X. Jin, "Research on user influence based on advanced LAD: taking Baidu Tieba as an example," in *Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 728–733, Jeju Island, Republic of Korea, April 2021.
- [31] H. Mao, M. Alizadeh, and I. Menache, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pp. 50–56, Atlanta, GA, USA, November 2016.
- [32] T. Xie, C. Zhang, and Z. Zhang, "Utilizing active sensor nodes in smart environments for optimal communication coverage," *IEEE Access*, vol. 7, pp. 11338–11348, 2018.
- [33] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- [34] L. Wu, Z. Yang, Q. Wang et al., "Fusing motion patterns and key visual information for semantic event recognition in basketball videos," *Neurocomputing*, vol. 413, pp. 217–229, 2020.
- [35] J. Rojas-Simón, Y. Ledeneva, and R. A. García-Hernández, "Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm," *Expert Systems with Applications*, vol. 167, Article ID 113827, 2021.
- [36] X. Fu, K. Zhang, C. Wang, and C. Fan, "Multiple player tracking in basketball court videos," *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 1811–1828, 2020.
- [37] L. Chen and W. Wang, "Analysis of technical features in basketball video based on deep learning algorithm," *Signal Processing: Image Communication*, vol. 83, Article ID 115786, 2020.
- [38] L. Wu, Z. Yang, and J. He, "Ontology-based global and collective motion patterns for event classification in basketball videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2178–2190, 2019.
- [39] K. Zhong, Y. Wang, and J. Pei, "Super efficiency SBM-DEA and neural network for performance evaluation," *Information Processing & Management*, vol. 58, no. 6, Article ID 102728, 2021.