

# Hiring Now: A Skill-Aware Multi-Attention Model for Job Posting Generation

Liting Liu<sup>1</sup>, Jie Liu<sup>2\*</sup>, Wenzheng Zhang<sup>2</sup>, Ziming Chi<sup>2</sup>, Wenxuan Shi<sup>1</sup>, Yalou Huang<sup>1</sup>

<sup>1</sup>College of Software, Nankai University, Tianjin, China

<sup>2</sup>College of Artificial Intelligence, Nankai University, Tianjin, China

{liu\_liting, wzzhang}@mail.nankai.edu.cn

zimingchi@163.com

{jliu, shiwx, huangyl}@nankai.edu.cn

## Abstract

Writing a good job posting is a critical step in the recruiting process, but the task is often more difficult than many people think. It is challenging to specify the level of education, experience, relevant skills per the company information and job description. To this end, we propose a novel task of Job Posting Generation (JPG) that is cast as a conditional text generation problem to generate job requirements according to the job descriptions. To deal with this task, we devise a data-driven global Skill-Aware Multi-Attention generation model, named SAMA. Specifically, to model the complex mapping relationships between input and output, we design a hierarchical decoder that we first label the job description with multiple skills, then we generate a complete text guided by the skill labels. At the same time, to exploit the prior knowledge about the skills, we further construct a skill knowledge graph to capture the global prior knowledge of skills and refine the generated results. The proposed approach is evaluated on real-world job posting data. Experimental results clearly demonstrate the effectiveness of the proposed method<sup>1</sup>.

## 1 Introduction

Writing high-quality job postings is the crucial first step to attract and filter the right talents in the recruiting process of human resource management. Given job descriptions and basic company information, the key to the job posting is to write job requirements, which requires to specify professional skills properly. Both too many or few requirements may lead to negative impacts on talent recruiting. Because of the extremely large number of job positions and varieties of professional skills, a lot of

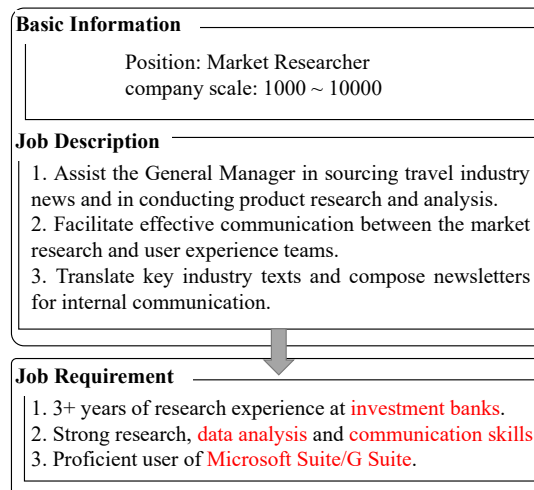


Figure 1: An example of automatic job posting.

companies have to pay much cost in this step to win in the war of talents.

To this end, we propose the task of Job Posting Generation (JPG) in this paper, and we cast it as a novel conditional text generation task that generates the job requirement paragraph. Exploiting the ubiquitous job posting data, we aim to automatically specify the level of necessary skills and generate fluent job requirements in a data-driven manner, as shown in Figure 1.

Although the JPG task is of great significance, the complexity of it poses several key challenges: 1) Generating job requirements needs to not only produce overall fluent text but also precisely organize the key content like skills and other information, which is very difficult to current neural systems. Especially, the long-text to long-text generation easily leads to information missing (Shen et al., 2019). 2) The key points of job descriptions and the skills of job requirements are complex many-to-many relations, which makes the mapping learning very difficult. 3) How to exploit the global information among the heterogeneous relations between basic company information and the professional

\*Corresponding Author

<sup>1</sup><https://github.com/NKU-IIPLab/SAMA>

skills across the whole dataset is of great importance to generate high-quality job requirements.

To address these challenges, we focus on the richness and accuracy of skills in generated job requirements and propose a global Skill-Aware Multi-Attention (SAMA) model for JPG task. Specifically, we devise a two-pass decoder to generate informative, accurate, and fluent job requirement paragraph. The first-pass decoder is to predict multiple skills according to the job description, which is a multi-label classification task (Zhang and Zhou, 2014). The second-pass decoder is to generate a complete text according to the predicted skill labels and the input text. Moreover, we build a skill knowledge graph to capture the global information in the whole job posting dataset in addition to the local information provided by the input. Through the skill knowledge graph, our model obtains the global prior knowledge to alleviate the misusing of skills. Extensive experiments are conducted to evaluate our model on real-world job posting data. The result demonstrates the effectiveness of the proposed method.

The main contributions of this paper can be summarized as follows:

- We propose a novel task of job posting generation that is defined as the conditional generation given a job description and basic company information to generate a job requirement.
- A data-driven generation approach SAMA is proposed to model the complex mapping relationships and generate informative and accurate job requirements.
- We build a real-world job posting dataset and conduct extensive experiments to validate the effectiveness and superiority of our proposed approach.

## 2 Data Description

We collect a job posting dataset from a famous Chinese online recruiting market, across a period of 19 months, ranging from 2019 to 2020. There are 107,616 job postings in total. After removing repetitive and too short job postings, 11,221 records are selected. This dataset is collected from 6 different industry domains. The detailed statistics of the dataset are illustrated in Table 1.

Considering the importance of the skills for JPG, we select 2000 records and manually tag the skills in these records. Then we train a word-level LSTM-

	training	validation	testing
Internet	2055	509	687
Consumer goods	1153	292	356
Real Estate	969	220	276
Finance	1477	366	463
Automobile	997	282	296
Medical	397	94	115

Table 1: The statistics of the dataset.

CRF model (Huang et al., 2015) to recognize the skills in the whole dataset.

We also keep the basic information, i.e., job position and company scale information, for the reason that they are the critical attributes of job postings that have impacts on the level of skills.

In order to capture the global prior knowledge of skills, we construct a skill knowledge graph according to the semantic relations of entities in the job postings. As shown in Figure 2, there are three types of entities, i.e., skill, company scale, and job position. The entities of skills are divided into two types, generic skills (denoted by G) and professional skills (denoted by P), according to the number of occurrences. The relation N.T.M. (need-to-master) exists between job position entity and skill entity. Besides, the relation IN exists between company scale entity and skill entity. For example, jobseeker who is seeking for a programmer position in a company of 10 to 100 people needs to master the professional skill C++, then there exist three triplets, (programmer, N.T.M., C++), ([10, 100], IN, C++) and (C++, type, P).

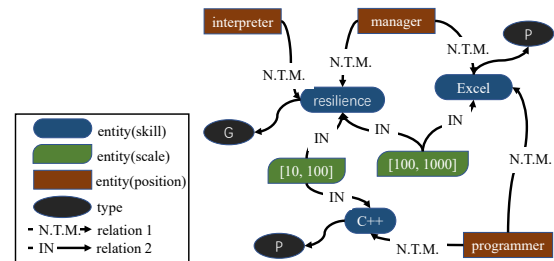


Figure 2: An example of the skill knowledge graph.

## 3 Approach

Let  $\mathcal{D} = \{(B_i, X_i, Y_i)\}_{i=1}^N$  denote the dataset, where  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$  is the word sequence of job description paragraph.  $Y_i = (y_{i,1}, y_{i,2}, \dots, y_{i,n})$  is the word sequence of job requirement paragraph,  $B_i = (b_i^p, b_i^s)$  is the basic information,  $b_i^p$  and  $b_i^s$  are job position and company scale information,  $N$  is the size of dataset,  $m$  and  $n$  are the lengths of sequence  $X_i$  and  $Y_i$ , respectively. The target of the JPG task is to estimate  $P(Y_i|X_i, B_i)$ , the conditional probability of a

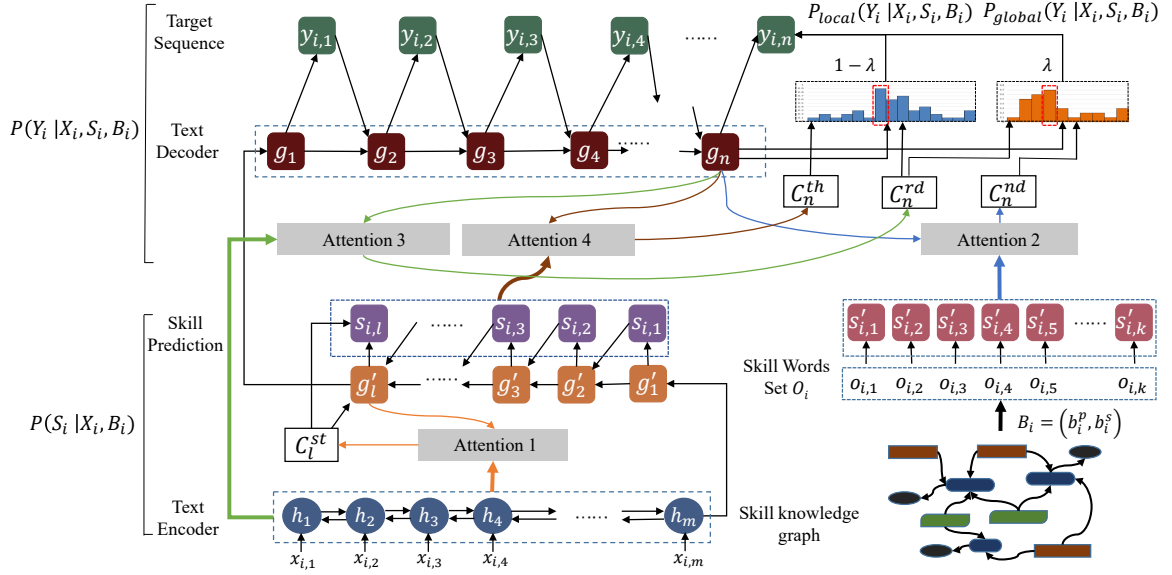


Figure 3: An illustration of the architecture of SAMA that consists of three parts, i.e., skill prediction part, skill refinement part, and job requirement generation part. The skills  $S_i$  are predicted given the job description. To consider the global prior knowledge of skills, the skill knowledge graph gives another set of skills  $O_i$ , which plays the role of refinement. Finally, SAMA fuses multiple attentions to generate the final job requirement paragraph  $Y_i$ .

job requirement  $Y_i$  given a job description  $X_i$  and basic information  $B_i$ .

To tackle the JPG task, we propose a global Skill-Aware Multi-Attention model, named SAMA. Figure 3 shows the overall architecture of SAMA.

Firstly, considering the importance of skill prediction in JPG, we decompose the probability  $P(Y_i|X_i, B_i)$  into a two-stage generation process, including skill prediction and job requirement paragraph generation:

$$P(Y_i|X_i, B_i) = P(Y_i|X_i, S_i, B_i)P(S_i|X_i, B_i), \quad (1)$$

where  $S_i = (s_{i,1}, s_{i,2}, \dots, s_{i,l})$  is a skill<sup>2</sup> word sequence of its corresponding job requirement,  $l$  is the length of  $S_i$ . Since  $S_i$  and  $B_i$  are conditionally independent given  $X_i$ , we can derive that  $P(S_i|X_i, B_i) = P(S_i|X_i)$ .

Secondly, for refining the skills, we leverage the global prior information by the skill knowledge graph  $G^s = (E_1, R, E_2)$  where  $E_1$  and  $E_2$  are the sets of head and tail entities and  $R$  is the set of relations. Given the basic information  $B_i$  and the skill knowledge graph  $G^s$ , we obtain a set of skills  $O_i = (o_{i,1}, o_{i,2}, \dots, o_{i,k})$ .

$$O_i = f(B_i, G^s), \quad (2)$$

where  $f$  is an invertible query function, which can ensure the one to one mapping relation between  $B_i$  and  $O_i$ .

<sup>2</sup>The details of how skills are extracted are described in Section 2.

Thirdly, to fuse the local and global information, the probability  $P(Y_i|X_i, S_i, B_i)$  during the text generation process is calculated as:

$$P(Y_i|X_i, S_i, B_i) = (1 - \lambda)P_{local}(Y_i|X_i, S_i, B_i) + \lambda P_{global}(Y_i|X_i, S_i, B_i), \quad (3)$$

where  $\lambda$  is a hyperparameter that adjusts the balance of two probabilities.

### 3.1 Job Description Encoder

The input job description word sequence  $X_i$  is first transformed into a sequence of word embeddings. To obtain the long-term dependency vector representation, we use a bi-directional LSTM (Schuster and Paliwal, 1997) as the text encoder. The input sequence is transformed into a hidden state sequence  $H = (h_1, h_2, \dots, h_m)$  by concatenating the representations of the forward and backward hidden states  $h_t = [\vec{h}_t, \overleftarrow{h}_{m-t+1}]$ . Specifically, the initiated encoder hidden state  $h_0$  is a zero vector, and the last encoder hidden state  $h_m$  is used for initiating the skill decoder.

### 3.2 Skill Prediction

Intuitively, the process of skill prediction is a Multi-Label Classification (MLC) task, which aims to assign multiple skills to each job description. To capture the correlations between skills, inspired by Yang et al. (2018), we view this MLC task as a sequence generation problem.

Formally, the skill decoder layer first takes the hidden state  $h_m$  of the encoder as input, then derive a context vector  $C^{st}$  by an attention mechanism (Luong et al., 2015) to help predict the skill labels.

$$\alpha_i^j = \frac{\exp(g_{j-1}^T W^1 h_i)}{\sum_{i'} \exp(g_{j-1}^T W^1 h_{i'})}; C_j^{st} = \sum_{i=1}^m \alpha_i^j h_i, \quad (4)$$

where  $W^1 \in \mathbb{R}^{d \times d}$  is trainable weight matrix,  $d$  is the hidden vector size. Inspired by Yuan et al. (2018), the job description is labelled with multiple skills by generating a skills sequence which joins the skills by delimiter <SEP> and has an unfixed number of skills (e.g., English <SEP> computer science <SEP> c++). The skill decoder is based on LSTM, whose hidden vector is computed by:

$$g'_t = LSTM(g'_{t-1}, C_t^{st}). \quad (5)$$

Specifically, the last skill decoder hidden state  $g'_t$  is used for initiating the text decoder. The skill sequence is finally obtained by a softmax classification over the vocabulary of skills,  $V_{skill}$ . In detail, a non-linear transformation is applied to form the skill decoder semantic representation  $I^{st}$ , and then compute the probability  $P(S_i|X_i, B_i)$  via:

$$I_j^{st} = \tanh(W^2[g'_j; C_j^{st}]) \\ P(s_{i,j}|X_i) = \text{softmax}_i(W^3 I_j^{st} + b^3), \quad (6)$$

where  $[\cdot; \cdot]$  is vector concatenation,  $W^2 \in \mathbb{R}^{d \times 2d}$ ,  $W^3 \in \mathbb{R}^{|V_{skill}| \times d}$  and  $b^3 \in \mathbb{R}^{|V_{skill}|}$  are parameters.

### 3.3 Skill Refinement

The process of skill prediction only considers the local information, which results in some misusing of skills. To refine the skill of the generated job requirement, the global information is taken into account by the skill knowledge graph.

The skill entities are divided into G and P as described in Section 2. Here, the basic assumption is that a generic skill appears more frequently than a professional skill among all the job postings, because the professional skill contains more domain characters. We use a hyperparameter  $\theta$  as a threshold to divide the skills entities.

Given the basic information  $B_i = (b_i^p, b_i^s)$ , the set of skills  $O_i$  is obtained from the skill knowledge graph by the query function  $f$ . In detail, firstly, we obtain the set of entities that have the ‘‘N.T.M.’’ relation with  $b_i^p$  and the set of entities who have the ‘‘IN’’ relation with  $b_i^s$ . Secondly, we get the

intersection of the sets obtained in the first step. Finally, we keep the entities whose types are P.

we embed  $O_i$  as  $S'_i = (s'_{i,1}, s'_{i,2}, \dots, s'_{i,k})$ , and linearly combine it as a skill graph context vector  $C_j^{nd}$  by an attention mechanism:

$$\tau_i^j = \frac{\exp(g_{j-1}^T W^4 s'_i)}{\sum_{i'} \exp(g_{j-1}^T W^4 s'_{i'})}; C_j^{nd} = \sum_{i=1}^k \tau_i^j s'_i, \quad (7)$$

where  $W^4 \in \mathbb{R}^{d \times d'}$  are parameters,  $d'$  is the dimensions of the word embeddings. Then a non-linear transformation is applied to form the graph skill semantic representation  $I^{nd}$ . The probability  $P_{global}(Y_i|X_i, S_i, B_i)$  from  $V_{skill}$  is computed via:

$$I_j^{nd} = \tanh(W^5[g_j; C_j^{nd}; C_j^{rd}]), \quad (8)$$

$$P_{global}(y_{i,j} = w|X_i, S_i, B_i) = \begin{cases} \text{softmax}_i(W^6 I_j^{nd} + b^6), & w \in O_i \\ 0, & w \notin O_i \end{cases}, \quad (9)$$

where  $g$  and  $C^{rd}$  will be introduced in next section,  $W^5 \in \mathbb{R}^{d \times (2d+d')}$ ,  $W^6 \in \mathbb{R}^{|V_{skill}| \times d}$ ,  $b^6 \in \mathbb{R}^{|V_{skill}|}$  are trainable parameters.

### 3.4 Job Requirement Generation

Job requirement generation fuses multiple attention mechanisms from three aspects, job descriptions, predicted skills and skills from skill knowledge graph. The text decoder, based on another LSTM, aims to generate final word sequence. The hidden vector of text decoder is computed by  $g_t = LSTM(e_{t-1}, g_{t-1})$ , where  $e_{t-1}$  is the word embedding of the final generated target word at time step  $t - 1$ . After obtaining  $g$ , a non-linear transformation is applied to form the text decoder semantic representation  $I^{rd}$ . The probability  $P_{local}(Y_i|X_i, S_i, B_i)$  is computed via:

$$I_j^{rd} = \tanh(W^7[e_{j-1}; g_j; C_j^{rd}; C_j^{th}]), \quad (10)$$

$$P_{local}(y_{i,j}|X_i, S_i, B_i) = \text{softmax}_i(W^8 I_j^{rd} + b^8), \quad (11)$$

where  $W^7 \in \mathbb{R}^{d \times 2(d+d')}$ ,  $W^8 \in \mathbb{R}^{|V_{text}| \times d}$ ,  $b^8 \in \mathbb{R}^{|V_{text}|}$  are parameters,  $V_{text}$  is the vocabulary of job requirement and  $V_{skill}$  is a subset of  $V_{text}$ , both  $C^{rd}$  and  $C^{th}$  are the context vectors generated by attention mechanisms. Specifically,  $C^{rd}$  is a context vector computed similar as  $C^{st}$  because they

directly take input sequence into account.

$$\beta_i^j = \frac{\exp(g_{j-1}^T W^9 h_i)}{\sum_{i'} \exp(g_{j-1}^T W^9 h_{i'})}; C_j^{rd} = \sum_{i=1}^m \beta_i^j h_i, \quad (12)$$

where  $W^9 \in \mathbb{R}^{d \times d}$ .

In addition, the skills  $S$  generated by skill decoder are fed into the text decoder to guide the generation process. To obtain  $C^{th}$ , another attention model is leveraged:

$$\gamma_i^j = \frac{\exp(g_{j-1}^T W^{10} s_i)}{\sum_{i'} \exp(g_{j-1}^T W^{10} s_{i'})}; C_j^{th} = \sum_{i=1}^l \gamma_i^j s_i, \quad (13)$$

where  $W^{10} \in \mathbb{R}^{d \times d'}$  are parameters.

The generation probability  $P(Y_i|X_i, S_i, B_i)$  is the weighted sum of  $P_{local}(Y_i|X_i, S_i, B_i)$  and  $P_{global}(Y_i|X_i, S_i, B_i)$  as in equation 3. As shown in equation 8 and equation 10, the vector  $C^{th}$  appears explicitly only in  $P_{local}$ , which implies that  $P_{local}$  puts emphasis on the skill prediction, i.e., the local information, while the vector  $C^{nd}$  appears explicitly only in  $P_{global}$ , which indicates that  $P_{global}$  focuses on the skills given by skill knowledge graph, i.e., the global prior knowledge.

In this way, SAMA considers not only the local information from the job description but also the global information from the skill knowledge graph.

### 3.5 Training and Inference

The loss function of the model has two parts, the negative log-likelihood of the silver<sup>3</sup> skill labels,  $L^S$ , and the gold<sup>4</sup> job requirement text,  $L^Y$ :

$$\begin{aligned} L^S &= - \sum_{i=1}^l \log P(S|X, B), \\ L^Y &= - \sum_{i=1}^n \log P(Y|X, S, B), \\ L &= L^S + \mu L^Y, \end{aligned} \quad (14)$$

where  $\mu$  is a hyperparameter, we give more weight to the loss of gold job requirement. During inference, the outputs of the skill decoder and the text decoder are predicted as follows:

$$\widehat{S} = \arg \max_S P(S|X, B), \quad (15)$$

<sup>3</sup>The skill labels are silver standard, because it was not created by an expert but extracted by a trained model.

<sup>4</sup>The job requirement text is gold standard, because it was written by human and put out online.

$$\widehat{Y} = \arg \max_Y P(Y|X, \widehat{S}, B). \quad (16)$$

For each stage, we obtain the best results by utilizing the greedy search at each step.

## 4 Experiments

In this section, we conduct experiments to verify the effectiveness of SAMA.

### 4.1 Experimental Setup

#### 4.1.1 Datasets

Job descriptions and job requirements are tokenized by PyLtp<sup>5</sup> word segmenter. Table 1 shows the split of the dataset. There are 468 position entities, 9 scale entities, 31,090 skill entities, and 310,413 relation edges in the skill knowledge graph. The vocabulary of job descriptions contains 14,189 words, the vocabulary of skills contains 3,523 words, and vocabulary the job requirements contains 18,612 words.

#### 4.1.2 Comparison Models

To achieve the comprehensive and comparative analysis of SAMA, we compared it with two kinds of representative models: the standard generation model and the hierarchical generation model.

- S2SA: Seq2Seq with attention (Luong et al., 2015) is a standard generation model.
- DelNet: Deliberation networks model (Xia et al., 2017) is a hierarchical generation model which has a two-pass decoder to generate and polish the same target sequence.
- VPN: Vocabulary pyramid networks (Liu et al., 2019) is a hierarchical generation model which has the multi-pass encoder and decoder to generate a multi-level target sequence.
- SAMA(w/o pred): SAMA(w/o pred) is a degraded model of SAMA that removes the process of skill prediction for the ablation test.
- SAMA(w/o graph): SAMA(w/o graph) is another degraded model of SAMA that removes the process of skill refinement.

#### 4.1.3 Network Configuration

In all models, we pretrain *word2vec* (Mikolov et al., 2013) in the job posting dataset. We set the word embedding dimension as 100 and the hidden vector size as 400 in both encoding and decoding. We set

<sup>5</sup><https://github.com/HIT-SCIR/pyltp>

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
S2SA	44.78	29.96	20.33	13.11	44.43	20.02	8.87	3.62
DelNet	37.10	25.35	18.28	12.62	44.21	19.29	8.42	3.08
VPN	34.15	23.26	16.90	11.68	40.16	16.82	6.96	2.63
SAMA(w/o pred)	44.70	31.59	23.09	16.32	45.87	22.78	11.25	5.75
SAMA(w/o graph)	45.49	31.89	23.32	16.40	45.93	22.85	11.37	5.84
SAMA	<b>46.15</b>	<b>32.44</b>	<b>23.77</b>	<b>16.83</b>	<b>46.37</b>	<b>23.27</b>	<b>12.17</b>	<b>6.16</b>

Table 2: Word overlap based metrics.

the maximum number of words in each sequence of skills and each job requirement as 30 and 150, respectively. Also, the weighted parameters  $\lambda$  and  $\mu$  are set as 0.5 and 1.4, respectively. The threshold  $\theta$  is set as 100. We apply dropout (Zaremba et al., 2014) at a rate of 0.3. Models are trained for 15 epochs with the Adam optimizer (Kingma and Ba, 2015), and the batch size is 5.

#### 4.1.4 Evaluation Metrics

To evaluate the performance of SAMA, we employ the following metrics:

**Word overlap based metrics:** To evaluate the overall text generation quality, we employ BLEU-N (Papineni et al., 2002) and ROUGE-N (Lin, 2004) as evaluation metrics, in which BLEU-N is a kind of precision-based metric and ROUGE-N is a kind of recall-based metric.

**Skill prediction metrics:** Since the correctness of generated skills is of great importance in JPG, we further evaluate the quality of skills in generated job requirements, using Precision, Recall, and F1 value. To achieve this, we extract skills in the ground truth and generated text by a matching method based on the skill vocabulary  $V_{skill}$ .

**Human-based evaluation:** Since it is difficult to measure the comprehensive quality of the generated texts, i.e., both fluency of the texts and accuracy of the skills, in addition to automatic metrics above, we conduct a subjective evaluation following. Three graduated student volunteers are asked to evaluate the generated paragraphs. We randomly sample 50 pieces of data from the testing set. The job requirements generated by different models are pooled and randomly shuffled for each volunteer. Each generated paragraph is evaluated as bad (irrelevant skills or disfluent sentence), normal (basic relevant skills and fluent sentence), or good (rich and relevant skills and fluent sentence).

## 4.2 Results and Analysis

### 4.2.1 Overall Performance

Table 2 shows the results of word overlap based metrics. In terms of BLEU-N and ROUGE-N,

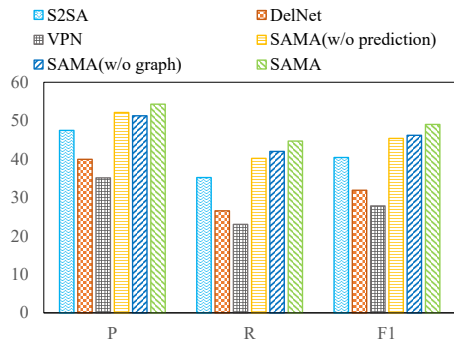


Figure 4: Skill prediction metrics.

SAMA performs the best in all word overlap based metrics, which suggests that our model obtains more overlapped words with the ground truth. SAMA(w/o graph) and SAMA(w/o pred) obtain competitive results, and both are significantly better than baselines, which demonstrates the effectiveness of skill prediction and prior knowledge of skills, respectively.

In addition to the overall metrics, Figure 4 further demonstrates the skill-level metrics. Figure 4 demonstrates that the job requirements generated by skill aware models (SAMA(w/o pred), SAMA(w/o graph), and SAMA) consist of more accurate and richer skills than those generated by the baselines (S2SA, DelNet, and VPN). Among them, SAMA achieves the best performance. Besides, SAMA(w/o graph) obtains a higher recall rate, which demonstrates that it can enrich the skill information effectively. SAMA(w/o pred) obtains a higher precision rate, which demonstrates that it can refine the skill information effectively.

### 4.2.2 Human-based Evaluation

Results of the human-based annotation are shown in Table 3. It can be seen that skill aware models obtain more relevant and informative results (good results) than the baselines, and SAMA obtains the most “good” results and the least “bad” results. The results are consistent with the automatic metric results. S2SA obtains the most “normal” results. This is because S2SA contains less rich and accurate skills in job requirements although with a good fluency. DelNet and VPN obtain a large percentage

Model	bad	normal	good	Kappa
S2SA	0.34	0.50	0.16	0.44
DelNet	0.48	0.34	0.18	0.41
VPN	0.56	0.32	0.12	0.38
SAMA(w/o pred)	0.28	0.42	0.30	0.42
SAMA(w/o graph)	0.26	0.42	0.32	0.43
SAMA	0.22	0.40	0.38	0.42

Table 3: Human-based evaluation results

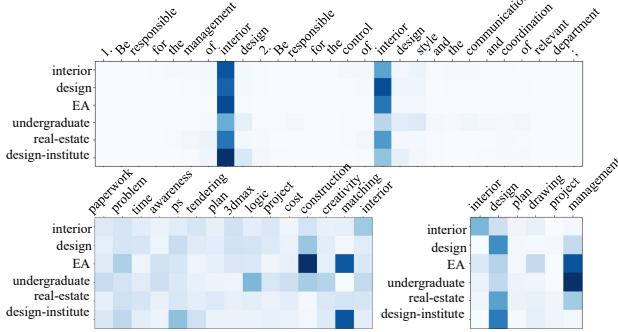


Figure 5: Visualization. The y-axes represent the skills of a generated job requirement, and the x-axes of the upper, the lower left and the lower right are the input job description  $X$ , the recommended skills  $O$  from the skill knowledge graph and skills  $S$  produced by the skill prediction, respectively.

of “bad” results mainly because of the repeated sentences. Besides, SAMA(w/o pred) and SAMA(w/o graph) are both much worse than SAMA on “good” results. This is because SAMA(w/o pred) misses some skills, and SAMA(w/o graph) misuses some skills. All models have the kappa scores around 0.4, indicating that evaluators reach high agreement.

#### 4.2.3 Visualization Analysis

When the model generates the target sequence, there exist differences in the contributions of different words. SAMA can synthetically select the most informative words by utilizing the three attention mechanisms. Figure 5<sup>6</sup> shows the visualization of three attention mechanisms. According to Figure 5, when SAMA generates the skill “EA (Environmental Art)”, it automatically assigns larger weights to more informative words in three sources, e.g., ‘interior’ of  $X$ , ‘interior, design, construction, matching’ of  $O$ , ‘interior, design, drawing, management’ of  $S$ . It shows that SAMA can consider the different contributions and capture the most informative words automatically from multiple sources.

#### 4.2.4 Case Study

To illustrate the difference in quality between SAMA and the compared models, we give an example of the generated text in Figure 6, where we

<sup>6</sup>Due to the space limitation, we intercept some texts.

Input:
1、负责完成公司下达的年度销售指标。2、将年度指标分解至季度、月度并加以执行。3、确保客户订单及时回款，确保无逾期、呆账等。4、渠道新客户开发及老客户的维护。
1. Responsible for completing the annual sales targets issued by the company. 2. Decompose annual indicators into quarters and months then implement them. 3. Ensure that orders are repaid timely and ensure no overdue or bad debts. 4. Develop new customer and maintain old customers.
Gold Output:
1、 <b>高中以上学历</b> ，具备一定的 <b>销售经验</b> 。2、有 <b>礼赠品团购</b> 渠道销售经验者优先。3、忠诚度高、服从管理、有 <b>团队协作精神</b> 。
1. <b>High school education above</b> , with some <b>sales experience</b> . 2. Sales experience in <b>gift group-buying</b> is preferred. 3. High loyalty, obedient management, and <b>teamwork spirit</b> .
SAMA Output:
1、 <b>高中以上学历</b> ，1年以上 <b>销售经验</b> ，有销售运营类管理更加的优先考虑；2、有 <b>礼赠品团购</b> 终端客户服务体系的工作经验、熟悉礼品销售者优先；3、有 <b>团队协作精神</b> ，能承受较大的工作压力。
1. <b>High school education above</b> , more than 1 year of <b>sales experience</b> , sales management is preferred; 2. Working experience in <b>gift group-buying</b> terminal customer service system, familiar with gift sales are preferred; 3. <b>Team spirit</b> , can bear high working pressure.
S2SA Output:
1、 <b>高中及以上学历</b> ，市场营销等相关专业；2、2年以上 <b>销售行业工作经验</b> ，有 <b>铝艺门窗或建材行业</b> 销售经验者优先。
1. <b>High school education or above</b> , marketing and other related majors; 2. More than 2 years <b>working experience in sales</b> , sales experience in <b>aluminum doors and windows or building materials industry</b> is preferred.

Figure 6: Case Study. We translate Chinese to English. **Skills in bold print** are the correct and accurate skills. The underlined skills are the correct but inaccurate skills. *The italic skills* are the incorrect skills.

compare SAMA with the strong baseline S2SA. As shown in Figure 6, SAMA captures all three aspects the same as ground truth, while S2SA misses the third aspect. Besides, in every aspect SAMA generates more correct and accurate skills, while S2SA obviously performs not good enough and generates inaccurate skills. Generally, the main consideration of job seekers is the skills they need to master, such as Python, English, and Go Language. Therefore, although S2SA generates some right words, like “preferred”, it does not increase the quality of the generated text because it generates inaccurate skills.

#### 4.2.5 Parameter Analysis

We show how the two key hyperparameters of SAMA,  $\lambda$  and  $\mu$ , influence the performance in Figure 7. The hyperparameter  $\lambda$  adjusts the balance of the probabilities between  $P_{local}$  and  $P_{global}$  and  $\mu$  adjusts the balance between two losses, the loss in skill prediction  $L^S$  and the loss in job requirements generation  $L^Y$ .

The value of hyperparameter  $\lambda$  varies from 0.1 to 0.9 and bigger value implies more global prior knowledge of skills. Figure 7 shows that the performance gets a peak when the  $\lambda$  increases. It is intuitive that prior knowledge can help generate

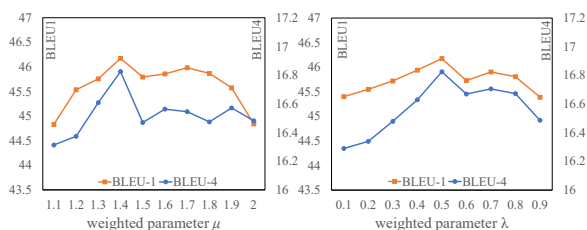


Figure 7: Parameter analysis.

accurate and rich skills. However, the too large value may sacrifice the fluency.

The value of hyperparameter  $\mu$  varies from 1.1 to 2.0. We give greater weight to the loss of job requirements generation for the reason that it is the target of the JPG task. As observed in Figure 7, a weight close to 1 may introduce noises from the skill labels. Besides, when the weight continuously increases close to 2, the model is incapable of fully considering the skill labels.

## 5 Related Work

The related works fall into two categories, human resource management and generation models.

### 5.1 Human Resource Management

Human Resource Management (HRM) is an appealing topic for applied researchers, and the recruitment is a key part of HRM. With the explosive growth of recruiting data, many studies focus on the efficient automatic HRM, e.g., person-organization fit, intelligent job interview, and job skill ranking. Lee and Brusilovsky (2007) designed a job recommender system with considering the preferences of both employers and candidates. Qin et al. (2019) proposed a personalized question recommender system for job interview to better interview the candidates. Naim et al. (2015) analyzed the videos of interview for quantifying verbal and nonverbal behaviors in the context of job interviews. Sun et al. (2019) studied the compatibility of person and organization. Xu et al. (2018) proposed a data driven approach for modeling the popularity of job skills. Besides, some augmented writing tools, such as Textio<sup>7</sup> and TapRecruit<sup>8</sup>, are developed to assist the HR to write job postings in the way that assuming a draft as input and then polishing the draft.

In this paper, we also consider improving the efficiency of HRM from the perspective of the job posting writing which is the crucial first step in the process of recruitment.

<sup>7</sup><https://textio.com/products/>

<sup>8</sup><https://taprecruit.co/>

## 5.2 Generation Models

Many practical applications are modeled as generation tasks such as keyword extraction, headline generation, and response generation. Many generation tasks are formulated as Seq2Seq learning problems. Plenty of studies focused on the optimization of the Seq2seq model. For example, Lopyrev (2015) trained a Seq2Seq model with attention for headlines generation task. Xing et al. (2017) incorporated topic information into Seq2Seq by a joint attention mechanism to generate informative responses for chatbots. Meng et al. (2017) applied a Seq2seq model with a copy mechanism to a keyword extraction task.

However, models without explicit modeling the sentence planning have a great limitation in generating complex argument structures depending on hierarchy. Dong and Lapata (2018) decomposed the semantic parsing process into sketch generation and details filled-in and proposed a structure-aware neural architecture. Zhang et al. (2019) formulated outline generation task as a hierarchical structured prediction problem and proposed HiStGen. Pudupully et al. (2019) proposed a two-stage model which incorporates content selection and planning, for the data-to-text generation task.

Similar to the above researches, we proposed a hierarchical generation model, namely SAMA, which first labels the job description with multiple skills and then generates the job requirement paragraph, to tackle the JPG task. Different from prior arts, SAMA considered the global information across the whole dataset to generate high quality job requirements.

## 6 Conclusion

In this paper, we proposed the job posting generation (JPG) task and formalized it to a conditional text generation problem. Besides, we proposed a novel model, SAMA, for this task. The merits of SAMA come from three aspects. Firstly, it decomposed the long text generation into two stages, including an MLC task and a multiple skills guided text generation task. Secondly, it considered both the local and the global information to generate accurate and rich skills. Last but not least, the learned mapping relationships can be applied to various downstream tasks, such as automatic resume, and person-job fit. Extensive experiments conducted on real-world job posting data demonstrated the effectiveness and superiority of SAMA.



## Acknowledgement

This research was supported by the National Natural Science Foundation of China under grant No. 61976119, the Natural Science Foundation of Tianjin under grant No. 18JCYBJC15800, and the Major Program of Science and Technology of Tianjin under grant No. 18ZXZNGX00310.

## References

- Li Dong and Mirella Lapata. 2018. [Coarse-to-fine decoding for neural semantic parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 731–742.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. [Bidirectional LSTM-CRF models for sequence tagging](#).
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *ICLR*.
- Danielle H Lee and Peter Brusilovsky. 2007. [Fighting information overflow with personalized comprehensive information access: A proactive job recommender](#). In *ICAS'07*, pages 21–21.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#).
- Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2019. [Vocabulary pyramid network: Multi-pass encoding and decoding with multi-level vocabularies for response generation](#). In *ACL*, pages 3774–3783.
- Konstantin Lopyrev. 2015. [Generating news headlines with recurrent neural networks](#).
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective approaches to attention-based neural machine translation](#). In *EMNLP*, pages 1412–1421.
- Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. [Deep keyphrase generation](#). In *ACL*, pages 582–592.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). In *1st International Conference on Learning Representations*.
- Iftekhar Naim, M Iftekhar Tanveer, Daniel Gildea, and Mohammed Ehsan Hoque. 2015. [Automated prediction and analysis of job interview performance: The role of what you say and how you say it](#). In *International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–6.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. [Data-to-text generation with content selection and planning](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6908–6915.
- Chuan Qin, Hengshu Zhu, Chen Zhu, Tong Xu, Fuzhen Zhuang, Chao Ma, Jingshuai Zhang, and Hui Xiong. 2019. [Duerquiz: A personalized question recommender system for intelligent job interview](#). In *KDD*, pages 2165–2173.
- Mike Schuster and Kuldip K. Paliwal. 1997. [Bidirectional recurrent neural networks](#). *IEEE Trans. Signal Processing*, 45(11):2673–2681.
- Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, and Lawrence Carin. 2019. [Hierarchically-structured variational autoencoders for long text generation](#).
- Ying Sun, Fuzhen Zhuang, Hengshu Zhu, Xin Song, Qing He, and Hui Xiong. 2019. [The impact of person-organization fit on talent management: A structure-aware convolutional neural network approach](#). In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1625–1633.
- Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. [Deliberation networks: Sequence generation beyond one-pass decoding](#). In *Advances in Neural Information Processing Systems*, pages 1784–1794.
- Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. [Topic aware neural response generation](#). In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3351–3357.
- Tong Xu, Hengshu Zhu, Chen Zhu, Pan Li, and Hui Xiong. 2018. [Measuring the popularity of job skills in recruitment market: A multi-criteria approach](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2572–2579.
- Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. [SGM: sequence generation model for multi-label classification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*.
- Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Daqing He, and Adam Trischler. 2018. [Generating diverse numbers of diverse keyphrases](#).
- Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. [Recurrent neural network regularization](#).
- Min-Ling Zhang and Zhi-Hua Zhou. 2014. [A review on multi-label learning algorithms](#). *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837.
- Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. [Outline generation: Understanding the inherent content structure of documents](#). In *SIGIR*, pages 745–754.