A Session-based Job Recommendation System Combining Area Knowledge and Interest Graph Neural Networks

Yusen Wang[a], Kaize Shi[a], Zhendong Niu[a,b,*]
[a] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[b] School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15260, USA
wangyusen@bit.edu.cn, kzshi@bit.edu.cn, zniu@bit.edu.cn

Abstract—**Online job boards become one of the central components of the modern recruitment industry. Existing systems are mainly focused on content analysis of resumes and job descriptions, so they heavily rely on the accuracy of semantic analysis and the coverage of content modeling, in which case they usually suffer from rigidity and the lack of implicit semantic relations. In recent years, session recommendation has attracted the attention of many researchers, as it can judge the user's interest preferences and recommend items based on the user's historical clicks. Most existing session-based recommendation systems are insufficient to obtain accurate user vectors in sessions and neglect complex transitions of items. We propose a novel method, Area Knowledge and Interest Graph Neural Networks(AIGNN). We add job area knowledge to job session recommendations, in which session sequences are modeled as graph-structured data, then GNN can capture complex transitions of items. Moreover, the attention mechanism is introduced to represent the user's interest. Experiments on real-world data set prove that the model we proposed better than other algorithms.**

Keywords-component; recommender system; session-based recommendation; GNN

I. INTRODUCTION

Now more and more candidates are looking for suitable jobs through the Internet. The traditional method of searching for keywords is inefficient, and users can only perform self-assessment based on recruitment information to determine whether they meet the post. This method cannot quickly find relevant recruitment data that matches candidates[1]. The recommendation system is a highly automated system that can efficiently recommend the items people need. The recommendation system has been applied in e-commerce[2], personalized advertising recommendation[3], e-learning recommendation[4]. In recent years, job recommendation systems have received increasing attention from researchers[5]. Job recommendation systems can quickly recommend suitable positions for job seekers.

Traditional job recommendation methods are mainly based on collaborative filtering[6], content-based approaches[7], the hybrid method[8]. Now deep learning method is also applied in job recommendation[9]. The core of session-based recommendation methods is to recommend new items to users based on the changing relationship among items. In job recommendation, people need to consider not only the position but also the workplace. Specific knowledge shows its advantages in multiple tasks[10][11]. In this paper, we introduce area knowledge and consider the regional characteristics in job recommendations to further improve the performance of the model. In real life, the user's behavior can be expressed as a sequence. The user's recent behavior sequence can be regarded as the user's current preferences, and the user's early historical session information implies the user's previous long-term interest preferences. Since the user's long-term preferences will change with time, so the current preference of the user cannot correctly reflect the real situation of the user to a certain extent. The previous job recommendation has weakened the influence of the difference between long-term and short-term interests on job seekers. This paper realizes more accurate job recommendations for users by considering the weight difference among long-term, short-term and global interests of job seekers. In this paper, we propose a job session recommendation model that combines area knowledge and interest graph neural networks(AIGNN).

II. RELATED WORK

A. Conventional Recommendation Methods

The matrix factorization is a common method in recommendation system. The basic objective is to factorize into two low-rank matrices according to a user-item rating matrix, and each matrix represents a latent factor of users or items[12]. This method is not suitable for the session-based recommendation system because the user preference is only provided by some positive clicks. The item-based neighborhood methods[13], this method is difficult to consider the order relationship among items. Markov decision processes[14], mainly learns the probability of state transition. The problem is that as items increase, it is very tough to model all possible click sequences.

B. Deep-learning-based Methods

Hidasi et al.[15] used RNN to form a deep neural network to predict the probability of the next clicked item in the session.

DOI reference number: 10.18293/SEKE2020-041

Tan et al.[16] improved the recurrent network model by using appropriate data augmentation techniques and taking into account temporal changes in user behavior. Tuan et al.[17] proposed using a 3D convolutional neural network to better model user-item interaction data and content features in sequence recommendation. Li et al.[18] pointed out that the previous method only considered the user's sequence performance, but the primary purpose of the user was not clearly emphasized. Hence, he proposed to adopt an attention mechanism on RNN to capture the sequential behavior characteristics and the main purpose of the user. Liu et al.[19] proposed a session recommendation model using MLP networks and current attention.

C. Graph Neural Network

Nowadays, graph neural networks can generate graph structure data representation. GNN [20] can represent the dependency among graph nodes. In recent years, GNN has some variants such as Gated graph neural network (GGNN)[21] and Graph Attention Network (GAT) [22]. Wu et al.[23] proposed using the GNN method to extract complex changes among items in session recommendation and achieved good results.

III. METHODS

In this section, we introduce the proposed AIGNN, in which the model is shown in Fig 1. We describe the AIGNN method thoroughly.

A. Notations

The goal of the session recommendation is to predict the most likely clicked item for the user's next step based on the user's previous session order. Here we define the letters that appear in AIGNN.

In session-based recommendation, let J $= \{j_1, j_2, j_{3,} \dots, j_m\}$ represents a set consisting of all unique items involved in all sessions. Each item contains the job knowledge clicked by the user and the area knowledge of the position. B $= \{b_1, b_2, b_{3,} \dots, b_m\}$ represents knowledge for each job. A $= \{a_1, a_2, a_{3,} \dots, a_m\}$ represents the area knowledge of each position. Fuse the information of A and B to form J, as shown in Eq. (1). An anonymous session sequence s can sort by timestamp to get list $s = [j_{s,1}, j_{s,2}, \dots, j_{s,n}]$, Where $j_{s,i} \in J$ represents the user's clicked item in session s. The goal of the model is to predict the next project $j_{s,n+1}$based on the previous clicked item of the user's. In each session, we calculate the probability $\hat{y}$ corresponding to each item and output the top-20 items as recommended items.

$$\mathrm{J} = \begin{bmatrix} \mathrm{j}_1 \\ \mathrm{j}_2 \\ \mathrm{j}_3 \\ \dots \\ \mathrm{j}_\mathrm{m} \end{bmatrix} = \begin{bmatrix} \mathrm{a}_1\ \mathrm{b}_1 \\ \mathrm{a}_2\ \mathrm{b}_2 \\ \mathrm{a}_3\ \mathrm{b}_3 \\ \dots \\ \mathrm{a}_\mathrm{m}\ \mathrm{b}_\mathrm{m} \end{bmatrix} \tag{1}$$

B. Session Graph

Since each item is a natural sequence, we construct an ordered session according to the click order of each item, and the ordered session sequence is converted into a session graph so that the GNN can process each session. We build each session s into a directed graph $\mathcal{G}_s = (j_s, \varepsilon_s)$. Each node in the session graph is an item $j_{s,i} \in J$. ε_s stands for all directed edge sets. $(j_{s,i-1}, j_{s,i}) \in \varepsilon_s$ $j_{s,i}$ represents the item clicked after clicking $j_{s,i-1}$. The weight of each edge is based on the occurrence of the edge divided by the outdegree of that edge's start node. We construct the vector of each session s according to the vector of each node item.

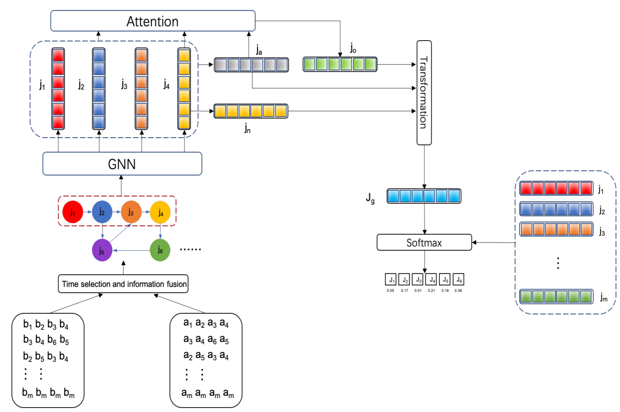

Figure 1. The overview of the proposed model.

C. Item Embedding

GNN can learn the complex relationship transformation of each node in the graph structure[20]. Gated graph neural network (GGNN) is a model based on GRU's spatial domain message passing[21]. It uses a similar principle of RNN to realize the transfer of information in a graph. In the field of job recommendation, in addition to considering the changing relationship among jobs, it is also necessary to consider the change of workplace. There are regional differences in job areas. Different regions focus on different types of jobs. Therefore, we have added the consideration of the regional factors to general session recommendations. Besides, we also consider the current, long-term, and global preferences of users to better analyze their behavior characteristics.

In this paper, we use GGNN to generate each item containing area knowledge. A connection matrix A_s is constructed to determine how each node communicates with each other, $A_s \in \mathbb{R}^{n \times 2n}$. Since each node of the calculation considers the bidirectional transfer relationship of the nodes, the connection matrix is $n \times 2n$ dimension as shown in Fig 2. $A_{s,i} \in \mathbb{R}^{1 \times 2n}$represents the connection matrix corresponding to each node in A_s. $H$is the weight, $z_{s,i}^t$ and $r_{s,i}^t$ are reset and update gates, σ(·) is the sigmoid function. The specific formula is as follows:

$$a_{s,i}^t = A_{s,i} : [j_1^{t-1}, \dots j_n^{t-1}]H + b \tag{2}$$

$$z_{s,i}^t = \sigma(W_z a_{s,i}^t + U_z j_i^{t-1}) \tag{3}$$

$$r_{s,i}^t = \sigma(W_r a_{s,i}^t + U_r j_i^{t-1}) \quad (4)$$

$$\widetilde{j_i^t} = \tanh(W_o a_{s,i}^t + U_o(r_{s,i}^t \odot j_i^{t-1})) \quad (5)$$

$$j_i^t = \left(1 - z_{s,i}^t\right) \odot j_i^{t-1} + z_{s,i}^t \odot \widetilde{j_i^t} \quad (6)$$

The gated graph neural network processes $\mathcal{G}_s$ for each session, and GGNN extracts the latent vectors of neighborhoods into the neural network. The reset gate is used to control the degree of ignoring the state information of the previous moment. The update gate is used to control the degree to which the state information of the previous moment is brought into the current state. Then we calculate the newly generated messages $\widetilde{j_i^t}$ based on the status of the previous, current, and reset gates. $\left(1 - z_{s,i}^t\right)$ is to select the forgotten information, j_i^t is the final updated node state.

D. Session Embedding

For a user's session representation, we consider the user's global representation, long-term interest representation, and current interest representation. The global preference takes the user's every click item. The long-term interest is obtained by averaging all the user's click information. The current interest is the user's last clicked item. The formula is as follows:

$$j_a = \frac{1}{m}\sum_{i=1}^{m} j_i \quad (7)$$

$$\beta_i = q^T \sigma(W_1 j_n + W_2 j_i + W_3 j_a + C) \quad (8)$$

$$j_o = \sum_{i=1}^{m} \beta_i j_i \quad (9)$$

$$J_g = W_4[j_n; j_a; j_o] \quad (10)$$

j_a is the average of the sum of users' long-term interest preferences according to their click items. $q \in \mathbb{R}^d$ and j_n is the user's last click as the user's current interest. j_o is the global embedding of the session graph $\mathcal{G}_s$. We use the attention mechanism to obtain the over-all preferences of users better.

Finally, we get the user's global preference, long-term preference, and current preference to obtain J_g through a linear change.

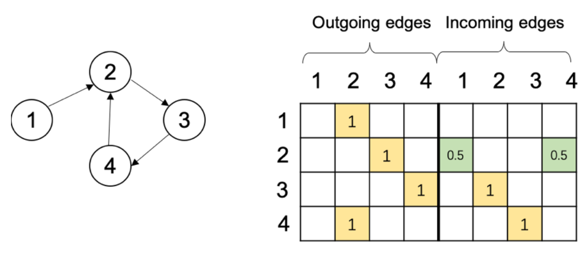

Figure 2. Connection matrix As representation

E. Recommendation

We calculate the integrated J_m and each item j_i to get the score $\hat{s_i}$ of each candidate item and then send the candidate score of each candidate item to softmax to calculate $\hat{y}$, which is the probability output of the next click in the click session. The formula is as follows:

$$\hat{s_i} = J_g^T j_i \quad (11)$$

$$\hat{y} = softmax(\hat{s}) \quad (12)$$

F. Training

For each session, we use cross-entropy as the training loss function. Calculate loss by prediction and the ground truth, as shown below:

$$\mathcal{L}(\hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i}) \quad (13)$$

IV. Experiments

A. Datasets

We verify the true validity of the AIGNN model recommendation in the job area. We use the real datasets CareerBuilder[1]. This is a real recruitment dataset, which contains user information, job information, etc. We selected 389,708 users, 10,913 job places, and a total of 315,105 jobs. We use 1023456 training sessions for training and 57858 test session data.

B. Baseline Methods

To evaluate the performance of the proposed method, we compare it with the following representative baselines: (1) POP: This is a simple baseline that ranks items according to their popularity measured by the number of interactions. (2) GRU4REC[15]: Session recommendation via RNN. (3) NARM[18]: The RNN with attention mechanism is used to capture the primary purpose and subsequent behavior of the user. (4) SR-GNN[23]: Session recommendation using gated graph neural network and attention mechanism.

C. Evaluation Metrics

Following metrics are used to evaluate compared methods.
Recall@20: It is widely used as a measure of prediction accuracy. It represents the proportion of correctly recommended items in the top-20 items.
MRR@20: It is the average of reciprocal ranks of the correctly-recommended items. The reciprocal rank is set to 0 when the rank exceeds 20.

V. Results and Analysis

The results with Recall@20 and MRR@20 on the recommendation performance are presented in Table I. We set the dimensionality of latent vectors d = 100, the learning rate is set to 0.001 and the L2 penalty is set to 10^{-5}.

TABLE I. PERFORMANCE COMPARISONS OF DIFFERENT METHODS ON THE SEQUENTIAL RECOMMENDATION TASK IN CAREERBUILDER

Method	Recall@20	MRR@20
POP	4.143	3.702

1 https://www.kaggle.com/c/job-recommendation/data

GRU4REC[15]	52.815	13.684
NARM[18]	63.181	17.572
SR-GNN[23]	68.324	19.729
AIGNN	**77.595**	**21.832**

For traditional algorithms such as POP, the performance is relatively poor. This simple model makes recommendations based on repeated co-occurring items or consecutive items. In NARM and GRU4REC, each user is explicitly modeled and represented by a separate sequence, ignoring the possible interaction among projects. SR-GNN considers the changing relationship among projects, without taking personalization modeling for users into account. We use the AIGNN, which combines the area knowledge with the job knowledge, and use the GNN method to extract the complex change relationship among projects more effectively. Besides, considering the current, long-term, and global preferences of users, the mechanism of attention can be more prepared to reflect the behavior characteristics of users, so it can achieve better results.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a Graph Neural Networks job session recommendation model based on area knowledge and interest. Considering the importance of job area knowledge for job recommendation system, The AIGNN method combines knowledge and job area knowledge and takes into account the user's long-term and current and global preferences. The changing relationship of items is extracted through GNN and calculated using softmax. Probability distribution selects the most suitable item. Comparison with other baseline methods on public datasets proves the effectiveness of our method.

Due to the session-based recommendation requires a large amount of user history to make recommendations, there is a problem of cold-start. In the future, it may be a research direction to add the mechanism of alleviating cold start into session-based job recommendation.

ACKNOWLEDGMENT

This research work is supported by the National Key R&D Program of China (Grant number: 2019YFB1406302).

REFERENCES

[1] K. Bradley and B. Smyth, “Personalized information ordering: a case study in online recruitment,” in Research and Development in Intelligent Systems XIX. Springer, 2003, pp. 279–292.

[2] O. Hinz and J. Eckert, “The impact of search and recommendation systems on sales in electronic commerce,” Business & Information Systems Engineering, vol. 2, no. 2, pp. 67–77, 2010.

[3] X. Liu and Y. Zhang, “A kind of personalized advertising recommendation method based on user-interest-behavior model,” in 2019 8th International Symposium on Next Generation Electronics (ISNE).IEEE, 2019, pp. 1–4.

[4] S. Wan and Z. Niu, “A hybrid e-learning recommendation approach based on learners’ influence propagation,”IEEE Transactions on Knowledge and Data Engineering, vol. 32,no. 5,pp. 827-840, 2020.

[5] M. Jiang, Y. Fang, H. Xie, J. Chong, and M. Meng, “User click prediction for personalized job recommendation,”World Wide Web,vol. 22, no. 1, pp. 325–345, 2019.

[6] M. Polato and F. Aiolli, “A preliminary study on a recommender system for the job recommendation challenge,” in Proceedings of the Recommender Systems Challenge, 2016, pp. 1–4.

[7] M. Bianchi, F. Cesaro, F. Ciceri, M. Dagrada, A. Gasparin, D. Grattarola, I. Inajjar, A. M. Metelli, and L. Cella, “Content-based approaches for cold-start job recommendations,” in Proceedings of the Recommender Systems Challenge 2017, 2017, pp. 1–5.

[8] Y. Lu, S. El Helou, and D. Gillet, “A recommender system for job seeking and recruiting website,” in Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 963–966.

[9] S. Benabderrahmane, N. Mellouli, and M. Lamolle, “On the predictive analysis of behavioral massive job data using embedded clustering and deep recurrent neural networks,”Knowledge-Based Systems,vol. 151, pp. 95–113, 2018.

[10] K. Shi, H. Lu, Y. Zhu, and Z. Niu, “Automatic generation of meteo-rological briefing by event knowledge guided summarization model,”Knowledge-Based Systems, vol. 192, p. 105379, 2020.

[11] K. Shi, C. Gong, H. Lu, Y. Zhu, and Z. Niu, “Wide-grained capsulenetwork with sentence-level feature to detect meteorological event insocial network,”Future Generation Computer Systems, vol. 102, pp.323–332, 2020.

[12] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in Advances in neural information processing systems, 2008,pp. 1257–1264.

[13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in Proceedings of the10th international conference on World Wide Web, 2001, pp. 285–295.

[14] G. Shani, D. Heckerman, and R. I. Brafman, “An MDP-based recommender system,”Journal of Machine Learning Research, vol. 6, no.Sep, pp. 1265–1295, 2005.

[15] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,”arXiv preprintarXiv:1511.06939, 2015.

[16] Y. K. Tan, X. Xu, and Y. Liu, “Improved recurrent neural networks for session-based recommendations,” in Proceedings of the 1st Work-shop on Deep Learning for Recommender Systems, 2016, pp. 17–22.

[17] T. X. Tuan and T. M. Phuong, “3D convolutional networks for session-based recommendation with content features,” in Proceeings of the Eleventh ACM Conference on Recommender Systems,2017, pp. 138–146.

[18] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, “Neural attentive session-based recommendation,” in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp.1419–1428.

[19] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, “Stamp: short-term attention/memory priority model for session-based recommendation,”in Proceedings of the 24th ACM SIGKDD International Conferenceon Knowledge Discovery & Data Mining, 2018, pp. 1831–183.

[20] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,”IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2008.

[21] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,”arXiv preprint arXiv:1511.05493, 2015.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero , P. Lio, and Y. Bengio, “Graph attention networks,” arXiv preprint arXiv:1710.10903,2017.

[23] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, “Session-based recommendation with graph neural networks,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 346–353.