Variable Interval Time Sequence Modeling for Career Trajectory Prediction: Deep Collaborative Perspective

Chao Wang[†]
Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China
Hefei, Anhui, China
wdyx2012@mail.ustc.edu.cn

Hengshu Zhu*
Baidu Talent Intelligence Center, Baidu Inc.
Beijing, China
zhuhengshu@baidu.com

Qiming Hao
Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China
Hefei, Anhui, China
hqm97@mail.ustc.edu.cn

Keli Xiao
Stony Brook University
Stony Brook, New York, USA
keli.xiao@stonybrook.edu

Hui Xiong*
Rutgers, The State University of New Jersey
Newark, New Jersey, USA
hxiong@rutgers.edu

ABSTRACT

In today's fast-evolving job market, the timely and effective understanding of the career trajectories of talents can help them quickly develop necessary skills and make the right career transitions at the right time. However, it is a non-trivial task for developing a successful career trajectory prediction method, which should have the abilities for finding the right timing for job-hopping, identifying the right companies, and matching the right positions for the candidates. While people have been trying to develop solutions for providing some of the above abilities, there is no total solution or complete framework to integrate all these abilities together. To this end, in this paper, we propose a unified time-aware career trajectory prediction framework, namely TACTP, which is capable of jointly providing the above three abilities for better understanding the career trajectories of talents. Along this line, we first exploit a hierarchical deep sequential modeling network for career embedding and extract latent talent factors from multiple networks, which are designed with different functions of handling related issues of the timing, companies, and positions for job-hopping. Then, we perform collaborative filtering for generating personalized predictions. Furthermore, we propose a temporal encoding mechanism to handle dynamic temporal information so that TACTP is capable of generating time-aware predictions by addressing the challenges for variable interval time sequence modeling. Finally, we have conducted extensive experiments on large-scale real-world data to evaluate TACTP against the state-of-the-art baselines, and the results show that TACTP has advantages over baselines on all targeted tasks for career trajectory prediction.

*Corresponding authors.
†This work was accomplished when the first author working as an intern in Baidu supervised by the second author.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Career trajectory prediction, Time sequence modeling, Temporal encoding

ACM Reference Format:
Chao Wang[†], Hengshu Zhu*, Qiming Hao, Keli Xiao, and Hui Xiong*. 2021. Variable Interval Time Sequence Modeling for Career Trajectory Prediction: Deep Collaborative Perspective. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449959

1 INTRODUCTION

Nowadays, job change has become a new normal for working life in the fast-paced business world. According to the US Bureau of Labor Statistics, 20-24% of Americans (i.e., more than 41 million people) search for new jobs every year since 2017 [1]. There are many benefits to understanding the career trajectory of talents from the perspectives of organizations, individuals, as well as the policymakers of labor and economics. For example, career trajectory analysis supports the human resource department in monitoring regional brain drain, making internal promotion decisions to motivate key talents, estimating the probability for a job seeker to accept the job offer through the hiring process, and many other meaningful tasks. Besides, from the perspective of talents, scheduling a satisfactory career trajectory from the abundant overload of job opportunities is usually a difficult decision to make, mainly due to the indecisiveness when facing high opportunity costs. Hence, career trajectory prediction results can benefit talents by guiding their career development paths. Also, career trajectory analysis is beneficial for the policymakers to find out the popular companies and corresponding cities when formulating policies and guidelines for attracting talents.

Regarding the career trajectory prediction, traditional methods are mainly based on empirical analysis [7, 41]. Recently, people

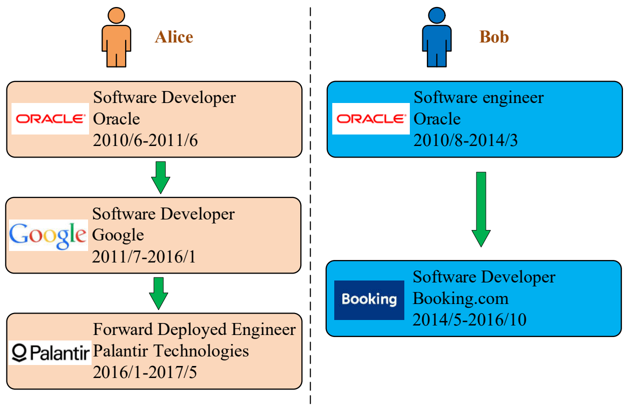


Figure 1: A motivating example of career path.

have witnessed the rapid development of professional social networks (PSNs) [38], such as LinkedIn and Glassdoor, supporting the over 200 billion recruitment market worldwide [1]. Corresponding job-related data enable people to seek effective machine learning solutions to the career trajectory prediction problem [24, 27, 30, 49]. For example, Li et al. [19] designed recurrent neural networks to predict talents' next companies and positions while Meng et al. [27] focused on forecasting potential companies and estimating the working durations for the next jobs by an attentive recurrent model. Based on convolutional neural networks, He et al. [10] predicted talents' future job positions, salaries, and company scales.

However, there are still many issues in existing models. First, existing solutions usually assume that the current job-hopping time is fixed and simply take historical job durations as a part of historical features [4, 10, 22, 27]. It is very difficult to capture the state changes for talents with variable intervals, and thus we should consider the interval between every two jobs as an important factor for sequential modeling. For example, in Figure 1, Alice and Bob have a similar starting point of careers. Alice only stayed a short period in Oracle while Bob chose to spend an extended period. Intuitively, they would have different choices for job transitions. An effective forecasting method should be capable of estimating the influences of different working durations, as well as the most likely time point for job-hopping. Importantly, we summarize three key aspects in career trajectory planning: the right timing for job-hopping, the right company for job application, and the right job position. None of the existing works aim to address all three prediction tasks in a unified manner. Moreover, due to the sparseness of individual career trajectory data, how to provide high-quality personalized predictions is also a long-standing challenge. Although some neighborhood based collaborative filtering (CF) methods [23, 51] have been developed, there still lacks a comprehensive and robust model to generate personalized prediction results.

To this end, in this paper, we propose a comprehensive hierarchical time-aware career trajectory prediction framework, namely TACTP, for jointly solving all the aforementioned problems. To the best of our knowledge, this is the first work to construct a joint model to predict the three key elements in career trajectory, i.e., timing, company, and position, simultaneously. Specifically, we propose a general temporal encoding mechanism to obtain latent time representations from original discrete time values. These time representations are incorporated in both sequential modeling and collaborative filtering stages to enhance our TACTP framework of recognizing and exploiting temporal information. Then, we employ the recurrent networks to map the heterogeneous feature inputs into three dynamic latent vectors representing talent-time, talent-company, and talent-position factors, respectively. Furthermore, we learn the dynamic latent company representations from deep collaborative networks and the latent position representations from Gaussian priors. As a result, our method could combine the latent company, position vectors with talent-company, talent-position, and time vectors to derive personalized time-aware predictions. Following this way, our TACTP framework can integrate the advantages of both sequential modeling and collaborative filtering to produce high-quality time-aware predictions according to different working durations of the current job. Finally, we conduct extensive experiments and evaluate TACTP by comparing it with state-of-the-art baselines on a large scale real-world dataset. The experimental results clearly demonstrate the effectiveness of TACTP in terms of all career trajectory prediction tasks.

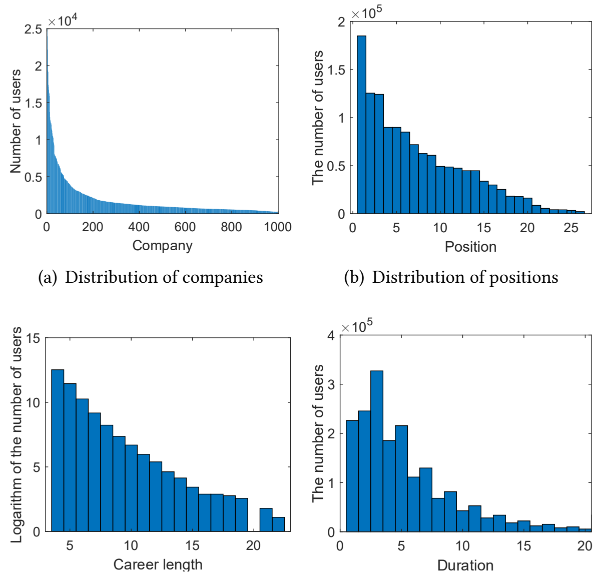
2 PRELIMINARIES

In this section, we first introduce the real-world dataset used in this study. Then we will formulate the prediction problem with the three correlated forecast targets, i.e., company, position and timing. Finally, we give an overview of our proposed TACTP Framework.

2.1 Data description

The real-world dataset in this paper was collected from LinkedIn, one of the most famous commercial professional social networks, which has served hundreds of millions of users to share their career experience and professional resumes. These resumes mainly contain two aspects of information, i.e., static and time-varying features. On one hand, static features are composed of user name, the number of user's social connections, and user's self-introduction text. On the other, time-varying features consist of the features related to companies and positions, such as company name, position name, and job duration of each working experience in their resumes.

However, only exploiting the above extracted information may be insufficient to support the modeling and inferring process for such a challenging prediction task from both user and item aspects. Therefore, we further adopt the following two measures to add more useful features to our study. First, we collected some other static company features from the open data sources as supplementary information, including company age, type, location, and description text. Second, we manufactured some relevant time-varying features by handling the original information into new forms. For example, we computed the working seniority for each user and the company size for each company at each time period as supplementary features. Also, we calculated the talent flow ratios for each company since talent flow analysis is much beneficial for monitoring companies' competitive advantages [42, 48]. Specifically, we counted the numbers of personal flow in/out/transfer records among companies every three years and then normalized these values. In this way,



(a) Distribution of companies (b) Distribution of positions

(c) Distribution of career lengths (d) Distribution of durations

Figure 2: Distributions of career trajectory records.

we can employ the previous talent flow in/out/transfer ratios as the time-varying features for the next time period.

The next challenge of this task is how to transform the massive raw data into standardized sequence forms. We believe the job transition process is a sequential modeling task and consider the career trajectory data as time series. However, in the real world, one talent may be employed by several companies at the same time. To solve this problem, we followed the treatment in [53] to find the major career path when there exist multiple parallel paths in talents' resumes. Specifically, if the absolute difference between the end time of the former job and the start time of the later job is less than a predefined time threshold, we regard this job transition as a valid record. Then we could construct a career tree if the former job has more than one valid later job. Finally, we only keep the longest career path in the career tree as the major career path. As for job positions, since the position names are provided by users, many positions in the raw data indeed have the same or similar meanings but totally different ways of expression. For normalizing these multifarious position names, we followed [53] to use an online API called MonkeyLearns[1] to classify the raw names into 26 job categories. Thus, in the following part of this paper, the job positions all refer to these 26 categories. Another notable thing is that a talent may change his/her position several times in one company. Since the career trajectory prediction task mainly focuses on the job-hopping behaviors among companies, job position transitions in one company are not the focus of our study. Thus, we only consider the first job position in each company for all the users. More details about the statistical information and pre-processing of the data can be found in Section 4.1.

[1]https://app.monkeylearn.com/main/classifiers.

To better understand the dataset properties, we present the distributions of career trajectory records from different views in Figure 2. Figure 2(a) and 2(b) show the distributions of the job records from company and position perspectives, respectively. Here we have sorted the companies and positions in descending order according to the number of users. From the company perspective, we can find that the long tail effect is quite obvious. Non-collaborative methods may be prone to focus on the top popular companies while neglecting the companies in the long tail. The distribution from the position perspective is also quite imbalanced. Therefore, it is important to learn independent representations for all companies and positions to solve the imbalanced data. Figure 2(c) presents the number of users of different career lengths. It is noticed that we have employed the logarithm to the number of users due to the imbalance data. Less than 5% users have been occupied by more than 6 companies. Figure 2(d) presents the counts of users of different working durations. A unit of abscissa in the figure means a half-year. It is interesting to observe the periodicity of durations, which shows the numbers of users in odd half-years are larger than even half-years except for the first year. Besides, more than 80% job experiences last less than 4 years, which also indicates the importance of career trajectory prediction.

2.2 Problem formulation

Based on the extracted career movement records and various features, here we introduce the problem formulation of the career trajectory prediction task.

The t-th job record of the i-th talent can be represented by the combination of three elements, i.e., company C_{it}, position P_{it} and duration D_{it}. Supposing the length of career path is T_i for talent i, we can denote the job sequence as $S_i = \{(C_{i1}, P_{i1}, D_{i0}), (C_{i2}, P_{i2}, D_{i1}), \dots, (C_{iT_i}, P_{iT_i}, D_{i,T_i-1})\}$. It is noticed that the working duration in the sequence refers to the $(t-1)$-th job. This is because we usually do not know how long the talents will stay in the current company until they move to the next company and update their resumes on PSN. We denote $D_{i0} = 0$ for the first job.

Similarly, the feature sequence for talent i can be denoted as $F_i = \{F_{i1}, F_{i2}, \dots, F_{iT_i}\}$. All the features can be classified into two types, i.e., static and time-varying features. Static features keep unchangeable throughout the whole career path. On the opposite, time-varying features would have different values over different time periods. For better usability, we first transform the original heterogeneous features into vectors using the preprocessing methods that will be introduced in Section 4.1. Then we concatenate the static and time-varying features to form the complete feature vector $F_{it} \in \mathbb{R}^r$ for each job record in the career path.

With the setup stated above, more formally, we define the prediction problem as follows:

Definition 2.1. (**Career Trajectory Prediction Problem**.) Given a sequence of job records S_i for talent i, along with the corresponding feature sequence F_i and length T_i, our goal is to predict the next job transition for talent i, including company C_{i,T_i+1}, position P_{i,T_i+1}, and duration D_{i,T_i}.

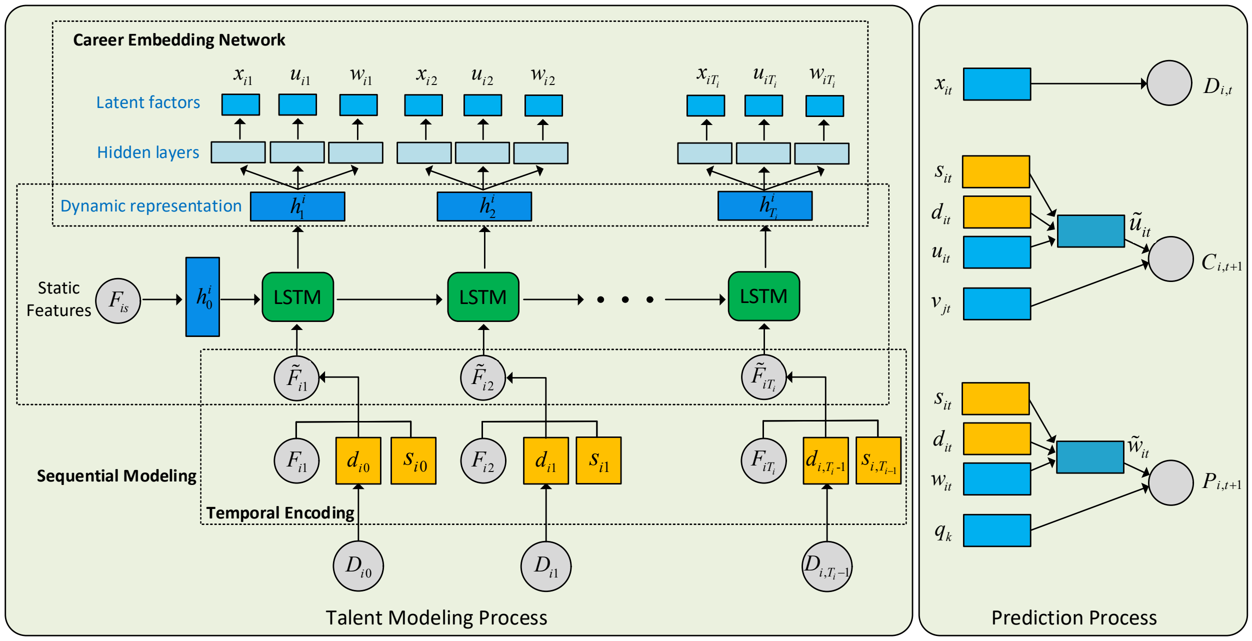


Figure 3: The network architectures of career modeling and prediction process in TACTP.

2.3 Solution overview

In this subsection, we will give an overview of our proposed TACTP Framework. As introduced in Section 2.2, TACTP aims to jointly make predictions on three different tasks: (1) find the most likely company for job-hopping; (2) identify the next job position; and (3) forecast the working duration for the current job. On one hand, the three prediction tasks are naturally closely related and mutually interacted in the real world. On the other hand, constructing a unified solution for the three tasks can help promote each other and obtain better understanding for the career trajectories. Consequently, it is improper to consider them separately. Along this line, we integrate the three prediction tasks in this paper. In TACTP, we employ the idea of latent factor based collaborative filtering [15] to factorize the job sequence records into latent vectors in a shared low-rank space. Specifically, TACTP could be divided into two stages, namely modeling stage and prediction stage. First, the modeling stage jointly constructs deep understandings for talent, company, position, and time factors. As illustrated in Figure 3, career modeling process can be further classified into three parts: 1. *Temporal encoding*, which transforms the raw time values into latent continuous space and combines them with input talents' features by an adaptive time perception layer; 2. *Sequential modeling*, which produces dynamic talent profiling information; and 3. *Embedding networks*, which maps the talent states to latent vectors. Prediction process aims to predict the three key elements (company, position, and timing) concurrently. Besides, we also learn the dynamic latent company representations from company embedding networks and latent position representations from Gaussian priors. Finally, we combine these latent vectors to produce the predictions for the three tasks in the prediction stage.

3 TECHNICAL DETAILS OF TACTP FRAMEWORK

In this section, we will present all the technical details of TACTP, including the modeling stage, the prediction stage, and the final comprehensive optimization objective.

3.1 Career sequential modeling

The goal of the career sequential modeling process is to model the individual career path of each talent. As shown in Figure 3, given the input features $F_i = \{F_{i1}, F_{i2}, \dots, F_{iT_i}\}$ and time vectors $\{d_{i0}, d_{i1}, \dots, d_{i,T_i-1}\}$, we want to learn the latent talent state h_t^i automatically for the i-th talent.

3.1.1 Temporal encoding. Time factor is quite important when talents make their choices on the job transition [47]. Generally, there are two types of temporal information playing influential roles in managing career paths. First is the working durations of past jobs, since historical working experiences will produce a sustainable influence on talents [34]. Career management researches have shown that historical working experience will produce a sustainable influence on their future career development [34]. Second, how long the talent stay in the current job also has a massive impact on career movement. As a result, we have to carefully consider the past time factor in the modeling stage and the current time factor in the prediction stage for generating time-aware predictions. Since the working durations are variable intervals, it is a great challenge to figure out their influences on job transitions.

To achieve this goal, we propose a general temporal encoding mechanism for capturing and exploiting the time factor in both modeling and prediction stage. In this paper, we focus on the career trajectory prediction tasks. However, this paradigm can be easily extended to other time sequential modeling problems with variable intervals. Specifically, we first transform the discrete time values into the continuous input embeddings for better usage in deep neural networks. Then, we design an adaptive time perception layer to transform the original time representations into individually customized time-aware representations.

Intuitively, the obtained latent representations from temporal encoding have to keep both the relative and absolute relations among original time values. For example, the representation of 5 years should be similar to 4 and 6 years while far away from 1 and 9 years. Here we introduce two implementations for deriving continuous time representations, namely manually specified approach and jointly learning approach.

Manually specified approach. To avoid increasing model complexity, manually specified approach chooses to utilize predetermined representations as the embedding results. In other words, the latent time representations are artificially designed and would not change during the model training. There are many workable choices for deciding the time representations [8]. In this paper, we draw the approach of positional encoding employed in the well-known sequential model, Transformers [43], to use sine and cosine functions with different frequencies for the vector d_{it}:

$$d_{it}^{(2j)} = \sin(D_{it}/10000^{2j/r}),$$
$$d_{it}^{(2j+1)} = \cos(D_{it}/10000^{2j/r}), \tag{1}$$

where $d_{it}^{(j)}$ means the j-th dimension of d_{it}. r is the dimension of d_{it}, which is the same as the feature vector F_{it}. Equation 1 has shown its effectiveness in NLP field for measuring spatial distance information. Here we can similarly exploit Equation 1 for measuring temporal distance information.

Jointly learning approach. Since the artificially designed representations may not accord with the practical relative and absolute relations of different time values, jointly learning approach chooses to learn the latent representations through the model learning. Specifically, we first randomly initialize the latent time vector for all the feasible time values in the dataset (or initialize with the values in manually specified approach). Thus according to the value of working duration D_{it}, we can assign the corresponding time vector d_{it} to the t-th working experience of talent i. Then the values of time vectors will be updated after each training batch. In this way, we can learn the relations among different time values by exploring real historical job records. We will further compare the performances of these two approaches in the experiments.

After getting the time representations, we need to combine them with the user information in the input feature vector F_{it} and time vector $d_{i,t-1}$ for producing time-aware input $\widetilde{F}_{it}$ at each time point. A direct way is to combine these two vectors by the concatenation operation, which is a common treatment for integrating information in deep learning methods:

$$\widetilde{F}_{it} = F_{it} \oplus d_{i,t-1}, \tag{2}$$

where $\oplus$ is the concatenation operation.

We can observe that the time vector $d_{i,t}$ in Equation 2 only depends on the working duration $D_{i,t}$. Thus, it would result in the invariant temporal information for a fixed input duration even when we change the user experience features. However, the timing

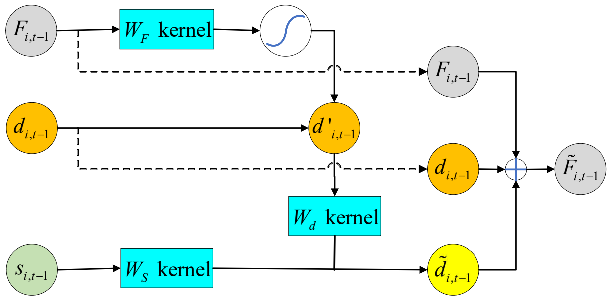


Figure 4: The time perception layer.

of job-hopping for a talent is largely depend on the current job status in practice. For example, talents in Internet companies tend to have more frequent career transitions and less working duration for each job. Moreover, even the same duration would naturally carry distinct meanings for talents with different working seniorities. Hence, in order to comprehensively measure the influence of temporal information with different job status and seniorities, we design an adaptive time perception layer to obtain individually customized time-aware representations.

Time perception layer. The time perception layer takes three input vectors F_{it}, $d_{i,t-1}$, and $s_{i,t-1}$, representing a talent's current job information, working duration, and seniority, respectively. Here the seniority before the t-th job record is defined as the sum of working durations of the past $t-1$ jobs for each talent. Then we can easily obtain the seniority vector $s_{i,t-1}$ by the introduced manually specified approach or jointly learning approach.

As shown in Figure 4, we first transform the input feature vector F_{it} through a nonlinear function and then let it interacts with the duration vector $d_{i,t-1}$ to analyze the influence from job status:

$$d'_{i,t-1} = f_F(W_F F_{it} + b_F) + d_{i,t-1}, \tag{3}$$

where f_F is a chosen nonlinearity, such as *tanh* function. W_F is a learned kernel and b_F is a bias term. Further, we project the influence of two time vectors $\widetilde{d}_{i,t-1}$ and $s_{i,t-1}$ onto a shared latent space by two linear functions:

$$\widetilde{d}_{i,t-1} = W_d d'_{i,t-1} + W_s s_{i,t-1}, \tag{4}$$

where W_d, W_s are learned kernels. Intuitively, the learned kernels would be trained to adapt the varying time scales and job experience of talents for mining the influence of temporal information on job-hopping. By comparison, $d_{i,t-1}$ only contains general time information while $\widetilde{d}_{i,t-1}$ integrates personal experiences with the duration vector. Finally, we combine the individually customized time vector $\widetilde{d}_{i,t-1}$ with the primary input features and duration vectors to form the final input vector:

$$\widetilde{F}_{it} = F_{it} \oplus d_{i,t-1} \oplus \widetilde{d}_{i,t-1}. \tag{5}$$

In this way, we can obtain the time-aware inputs, which reflect the influence of time factor in the modeling stage. We found the temporal encoding mechanism performs well across the three targeted prediction tasks. A similar temporal encoding process will also be used in the prediction stage with a little modification. We will discuss this later in Section 3.4.

3.1.2 Sequential modeling. With the input vector sequence described above, we then incorporate recurrent networks to track the dynamics of talent states. We can apply many workable network architectures to TACTP framework, such as Recurrent Neural Network (RNN) [12], Long Short-Term Memory (LSTM) [9] and Gated Recurrent Unit (GRU) [5].

Let us take LSTM as an example to introduce the modeling process. As introduced in section 2.2, each input feature F_{it} is the concatenation of static and time-varying features. The static features are all the same for $\forall t$. Thus given the static features F_{is} of talent i, we first employ a single hidden layer to get the initial hidden cell h_0^i of the recurrent network. Then given the time-aware input feature $\widetilde{F}_{it}$ for the t-th job, $(t-1)$-th neural cell c_{t-1}^i and $(t-1)$-th hidden state h_{t-1}^i, we can obtain the t-th hidden state by

$$h_t^i = LSTM(\widetilde{F}_{it}, c_{t-1}^i, h_{t-1}^i). \tag{6}$$

Notice that h_t^i is not only decided by the current state of talent i, but also the historical working experience. When using RNN or GRU for TACTP framework, the modeling process is analogous.

Finally, we adopt the dropout layer to randomly drop out the state h_t^i. In this way, the networks would receive different incomplete inputs for follow-up tasks in each training epoch. Thus, the dropout strategy can improve the robustness and generality of TACTP.

3.2 Career embedding networks

Since the obtained latent talent vectors are synthetical representations, we need to further specify the talent states from different perspectives. Specifically, there are three primary perspectives related to career trajectory prediction problems, i.e., company, position, and time.

From the company perspective, we can use an embedding network to transform the talent representation h_t^i into the talent-company vector u_{it}. We choose the multi-layer perception (MLP) networks as the embedding networks. Thus, we have:

$$\begin{aligned} g_1 &= f_1(W_1 h_t^i + b_1), \\ g_t &= f_t(W_t g_{t-1} + b_t), \qquad t \in [2, n-1], \\ u_{it} &= f_n(W_n g_{n-1} + b_n), \end{aligned} \tag{7}$$

where g_t is the t-th hidden layer with weight matrix W_t and bias term b_t. For the activation function $f_t(\cdot)$, we employ the *sigmoid* function for the first $n-1$ layers and the *tanh* function for the last layer. Usually in practice, 2-layer MLP has been good enough for generating high-quality embeddings.

Here, we let the embedding networks at different time points share the same weights. This is because we assume the talent representations are independent of time so that similar talent representations would result in similar talent-company vectors at any time point. Such treatment also helps reduce model complexity and prevent over-fitting. Similarly to talent-company vector, we also gain the latent talent-position vector w_{it} and talent-time vector x_{it} from position and time perspectives, respectively. The network architectures are the same as talent-company embedding network and the only difference is the parameters of embedding networks.

3.3 Company and position modeling

After modeling talents, we then need to construct the latent company and position vectors for utilizing collaborative filtering.

The latent talent vectors are not only influenced by the current job but also the past working experience. Consequently, we build a sequential modeling process for users. On the contrary, since company properties can be essentially represented by the current state without the need for historical states, we can directly use a shared embedding network to transform the features of the j-th company into latent company vector v_{jt} for all the time periods. Considering company features contain both static and time-varying features about the company, we are able to learn the time-aware representations in different periods. The network architectures are similar to Equation 7.

Different from the company perspective, the states of positions are much more steady and usually not vary with time. Thus we can employ time-independent latent position vectors for all time periods. In detail, we denote q_k as the latent position vector and then the prior probability over q_k is assumed to be the normal distribution as follows:

$$p(q_k) \sim \mathcal{N}(0, \lambda^{-1} I), \tag{8}$$

where λ is the regularization parameter. q_k will be updated by the gradients after each training batch.

3.4 Prediction stage

In the prediction stage, we finally combine the above discussed latent factors to produce the final prediction results for timing, company, and position, respectively.

For predicting the working duration of the current job, we can use a single hidden layer to transform x_{it} into the prediction $\hat{d}_{i,t}$, which is equal to the Logistic Regression. Here we suggest normalizing the original time values into the range $(0, 1)$ to make the model more robust. Thus the loss function for working duration prediction can be given by:

$$L_d = \sum_{i,t} \frac{1}{2} (\hat{d}_{i,t} - d_{i,t})^2. \tag{9}$$

For predicting the next company of talents, we first need to inject temporal information of the current job into talent-company vectors, since the time factor is an important decisive force for career movement. We also use 1-layer MLP $G(\cdot)$ to transform the time vector into the same space of talent vectors :

$$\begin{aligned} G(d_{it}) &= W d_{it} + b, \\ G(s_{it}) &= W s_{it} + b, \end{aligned} \tag{10}$$

where W is the weight matrix and b is the bias term. Then we can adopt the time perception layer to obtain the time-aware talent vector $\widetilde{u}_{it}$. There are some differences between the time perception layer in modeling stage and prediction stage. Firstly, the three input vectors $u_{it}, G(d_{it})$, and $G(s_{it})$ in prediction stage are in the low-rank latent space while $F_{it}, d_{i,t-1}$, and $s_{i,t-1}$ in modeling stage are in the feature space. Besides, in order to ensure the consistency in the dimension of latent space, we choose to add the concatenation of two latent time vectors to the talent vector u_{it}:

$$\widetilde{G}(d_{it}) = W_{gd}(f_u(W_u u_{it}) + G(d_{it})) + W_{gs}G(s_{it}),$$
$$\widetilde{u}_{it} = u_{it} + G(d_{it}) \oplus \widetilde{G}(d_{it}), \tag{11}$$

where the dimensions of both $G(d_{it})$ and $\widetilde{G}(d_{it})$ are equal to half of the dimension of u_{it}.

In this way, the obtained vector $\widetilde{u}_{it}$ would contain both the talent interest and temporal information. Notice that in the training process, we input the time vector d_{it} according to the real working duration D_{it}. Differently, in the prediction process, we can input varying duration values instead of the real working duration to gain different time-aware talent vectors. This is especially important when we do not know the ground truth of the duration of current job. A natural approach is to input our predicted working duration. Talents can also assign the user-specific working durations by themselves in practice.

Afterwards, we adopt collaborative filtering to get the final job transition probability $y_{i,t+1}$:

$$y_{i,t+1} = softmax(v_t^T \widetilde{u}_{it}), \tag{12}$$

where n is the number of companies and $v_t = \{v_{1t}, v_{2t}, ..., v_{nt}\}$ is the latent company matrix for the time point of the t-th job. $softmax(\cdot)$ is the softmax function. Hence the j-th dimension of $y_{i,t+1}$ represents the job transition probability to the j-th company.

Let $o_{it} \in \mathbb{R}^n$ denote the one-hot embedding of company C_{it}. Thus the C_{it}-th dimension of o_{it} is equal to 1 and otherwise 0. The loss function for next company prediction can be given by the cross-entropy form:

$$L_c = -\sum_{i,t} o_{i,t+1} \log(y_{i,t+1}). \tag{13}$$

For predicting the next position of talents, the prediction process is the same as predicting the next company. Thus we can similarly calculate the loss function L_p for the next position prediction.

We also add the regularization term to prevent model from overfitting problem:

$$L_r = \sum_{i,t,j,k} \frac{1}{2}(\|u_{it}\|^2 + \|w_{it}\|^2 + \|v_{jt}\|^2 + \|q_k\|^2). \tag{14}$$

Finally, we have the whole objective function as follows:

$$L = L_c + \alpha L_d + \beta L_p + \lambda L_r, \tag{15}$$

where α, β and λ are hyper-parameters for balancing the different parts in the loss function.

4 EXPERIMENTS

In this section, we will demonstrate the effectiveness of our proposed TACTP framework from the following aspects: (1) the overall prediction performance compared with state-of-the-art baselines on the three targeted prediction tasks; (2) the analysis on temporal encoding; (3) the model robustness evaluation; and (4) the analysis on latent company and position vectors.

Table 1: The statistical information of the dataset.

The number of total job records	1,872,624
The number of users	414, 266
The number of companies	1, 002
The number of positions	26
The mean value of the lengths of career paths	4.52

Table 2: The description of the features in our dataset.

Feature level	Feature Type	Category	Feature
User	Static	Categorical	User ID
		Numerical	Number of social connections
		Text	Self-introduction
	Time-varying	Numerical	working seniority
Company	Static	Categorical	Company ID Company age Company type Company location
		Text	Company description
	Time-varying	Categorical	Company size
		Numerical	Job duration Company flow in ratio Company flow out ratio Company flow transfer ratio
Position	Static	Categorical	Position ID

4.1 Data pre-processing

In this subsection, we introduce how to standardize the input features. The description of all the utilized features is given in Table 2. All the features could be classified into three categories (categorical, numerical, and textual data) according to the data forms. For categorical features, we first used one-hot encoding to obtain the vector representations. Then we further employed a single layer MLP to reduce the high-dimension features. For numerical time values, following the settings in [27], we set the time window as a half year and then segmented the time periods. In this way, the value of working durations and working seniorities could be presented as integers. For instance, 3 years and 9 months would be transformed into time value 8 since there are 8 half-years. We set the maximum time value as 21 for the working duration, which means all the working durations larger than 10 years were classified into one category. Then we used the introduced temporal encoding mechanism to further obtain the final time embedding. Meanwhile, we also utilized a single layer MLP to reduce the dimensions of numerical talent flow in/out/transfer ratios. Lastly, for textual data such as self-introduction and company description, we adopt the well-known text processing method, wold2vec [35], to transform the raw text into vectors.

4.2 Experimental settings

Dataset. In our dataset, the time span of the career path data ranges from 1988.1 to 2018.11. We first remove the companies with very few job records. Then we filtered out the users with less than four job records. After data filtering, the statistical information of the dataset is presented in Table 1. Specifically, in our experiments, we randomly sampled 80%/10%/10% users and their career paths to construct the training/validation/test set. In this way, we randomly

Table 3: The overall performance for next company prediction, next position prediction and current working duration prediction tasks. ("–" means the method is not suitable for this task.)

	Company				Position				Duration	
Methods	Acc@1	Acc@15	Acc@30	MRR	Acc@1	Acc@2	Acc@3	MRR	MAE	RMSE
PP	–	–	–	–	–	–	–	–	3.190	4.296
MHP	–	–	–	–	–	–	–	–	3.267	4.715
GBDT	0.022	0.196	0.320	0.075	0.212	0.327	0.413	0.369	3.992	5.471
RF	0.045	0.285	0.393	0.107	0.262	0.428	0.531	0.445	3.303	5.146
LR	0.055	0.317	0.425	0.123	0.341	0.494	0.596	0.511	3.226	4.994
CRF	0.058	0.336	0.453	0.129	0.344	0.459	0.537	0.491	3.277	5.091
CTMC	0.060	0.336	0.457	0.089	0.342	0.462	0.542	0.492	4.225	5.938
HCPNN	0.076	0.408	0.540	0.159	0.352	0.523	0.630,	0.528	–	–
NEMO	0.124	0.507	0.636	0.224	0.392	0.553	0.653,	0.559	–	–
TACTP (RNN)	0.104	0.481	0.612	0.200	0.390	0.551	0.652,	0.558	2.806	3.744
TACTP (GRU)	0.135	0.535	0.657	0.239	0.400	0.562	0.663,	0.555	**2.771**	3.752
TACTP-P (LSTM)	0.141	0.545	0.667	0.248	0.396	0.559	0.660,	0.564	2.780	3.761
TACTP (LSTM)	**0.150**	**0.560**	**0.680**	**0.258**	**0.401**	**0.564**	**0.664**	**0.568**	2.774	**3.732**

split each dataset five times and reported all the results by mean values. For each user, we validated the prediction performance of three tasks on every job record except for the first job.

Baseline approaches. To verify the effectiveness of TACTP, we compare it with some state-of-the-art baseline methods. Specifically, non-sequential models contain Logistic Regression (LR), Random Forest (RF), and Gradient Boosting Decision Tree (GBDT). Sequential models contain Conditional Random Field (CRF) [16], Continuous Time Markov Chain (CTMC) [2], HCPNN [27] and NEMO [19]. HCPNN and NEMO are the most advanced and relevant career trajectory prediction methods, which are both based on LSTM models. The original HCPNN model cannot predict the positions of talents. We modify HCPNN by replacing the input companies with positions and deleting the position embeddings in original model. Then we can train a new modified HCPNN model to predict the next positions. For predicting the timing, baselines also include the stochastic time series models, such as Poisson Process (PP) [14] and Multi-variable Hawkes Process (MHP) [26]. TACTP (RNN), TACTP (GRU), and TACTP (LSTM) are the three different implementations of TACTP framework using RNN, GRU, and LSTM as the recurrent network architectures, respectively. TACTP-P (LSTM) is a variant of TACTP (LSTM) where we input the predicted working durations instead of the real working durations in the prediction stage.

Evaluation metrics. To evaluate the performance of next company and position predictions, we adopted two widely used evaluation metrics, i.e., accuracy@K (Acc@K) and mean reciprocal rank (MRR). Acc@K counts the ratio that correctly predicted results are in talents' top-K items. Specifically, we calculated it by Acc@$K = \frac{1}{N}\sum_{l=1}^{N} I(r(l) \leq K)$, where $r(l)$ denotes the rank of the l-th predicted item and N is the number of predicted items. $I(\cdot)$ is the indicator function. Here we chose $K = 1, 15, 30$ for company prediction and $K = 1, 2, 3$ for position prediction. MRR measures the rank of the prediction items, i.e., MRR$=\frac{1}{N}\sum_{l=1}^{N}\frac{1}{r(l)}$. For duration prediction, we adopted two widely used evaluation metrics, mean absolute error (MAE) and root mean square error (RMSE). Generally, the larger the values of Acc@K, MRR are, and the smaller the values of MAE, RMSE are, the better results we have.

Parameter settings. In our experiments, the dimensions of all the latent talent-company, talent-position, company, and position vectors were set as 150. Accordingly, we chose talent-company, talent-position and company embedding networks in TACTP as 2-layer MLPs with dimensions 100×150. Meanwhile, the dimension of talent-time vector was set as 50 and the time embedding network was set as a 2-layer MLP with dimensions 50×50. Besides, we chose the jointly learning approach for combining input features with time vectors in the overall prediction performance comparisons. Then we tuned the values of hyper-parameters α in $[1, 2, ..., 10]$, β in $[0.1, 0.2, ..., 1]$, and λ in $[0.0, 0.01, ..., 0.1]$. The dropout ratio was set as 0.99. Finally, we performed Adam algorithm for optimization and tuned the learning rate from 0.0001 to 0.01. For all the baseline methods, we use the grid search to explore the parameters. Particularly, for the two LSTM based models, HCPNN and NEMO, we adopt the exactly same sizes with our TACTP model for their LSTM layers. Besides, for these two LSTM based models, we concatenated the time vectors with other feature embeddings, such as company and position vectors, to construct the network inputs. Here the time vectors were obtained just the same as the introduced jointly learning approach in our TACTP framework. Thus we can compare them with our models fairly.

4.3 Prediction performance comparison

We want to validate our proposed method on the three career trajectory prediction tasks: providing the right timing for job-hopping, identifying the right company for a job application, and matching the right position for the candidate. The overall prediction performance for three tasks is shown in Table 3.

First, for company prediction, it can be easily observed from Table 3 that TACTP significantly outperforms all the baselines, owing to the integration of sequential modeling and collaborative filtering. Specifically, TACTP (LSTM) outperforms the best baseline, NEMO, by the relative boost of 20.97%, 10.45%, 6.92%, and 15.18% for the metric ACC@1, ACC@15, ACC@30, and MRR, respectively. By comparison, HCPNN and NEMO are also LSTM based models, but they purely consider time as one of the fixed features and adopt a

Table 4: The performance of different ways for handling temporal information in TACTP framework.

	Company		Position		Duration
	Acc@1	MRR	Acc@1	MRR	MAE
TACTP-N	0.117	0.217	0.390	0.552	2.788
TACTP-C (M)	0.123	0.223	0.394	0.562	2.785
TACTP-O (M)	0.127	0.230	0.396	0.563	2.786
TACTP (M)	0.134	0.240	0.398	0.566	2.781
TACTP-C (J)	0.134	0.235	0.397	0.564	2.775
TACTP-O (J)	0.138	0.241	0.392	0.560	2.777
TACTP (J)	0.150	0.258	0.401	0.568	2.774

non-collaborative way for prediction. Differently, TACTP produces personalized time-aware prediction results and thus, achieves a large improvement to them in the variable interval time sequence tasks. Besides, we can observe that sequential models always have stronger modeling ability and perform better than non-sequential models, which demonstrates the necessity of sequential modeling again. As for the three implementations of TACTP, we can find that TACTP (LSTM) achieves the best result while TACTP (RNN) performs not well. This may be because LSTM model can largely alleviate the gradient vanish problem and is more suitable for variable interval time sequences than RNN and GRU. Finally, TACTP-P (LSTM) also produces similar results to TACTP (LSTM), which demonstrates that with the predicted working durations, our TACTP framework is still able to achieve comparable performance. In practice, we can further employ TACTP framework for handling varying interval inputs.

For position prediction, as shown in Table 3, TACTP (LSTM) achieves the best performance against all the baseline methods. Specifically, TACTP (LSTM) outperforms the best baseline, NEMO, by the relative boost of 2.30%, 1.99%, 1.68% and 1.61% for the metric ACC@1, ACC@15, ACC@30 and MRR, respectively. Similarly to the company prediction task, TACTP (LSTM) performs better than TACTP (RNN) and TACTP (GRU). We can observe that the absolute values of Acc and MRR in position prediction tasks are much larger than the company prediction task. This is because the number of companies is much larger than the number of positions. Thus it is much more difficult to provide precise predictions for companies than positions.

Lastly, for current job duration prediction, we can observe that our proposed TACTP models achieve the best performance in both evaluation metrics. Specifically, TACTP (LSTM) outperforms the best baseline, MHP, by the relative boost of 17.90% and 26.34% for the metric MAE and RMSE, respectively. We can find that the stochastic time series models, (PP and MHP) both have better performance than the other baseline approaches. However, with the help of deep sequential modeling and understanding of talents, TACTP can outperform them with a large margin. Different from company and position prediction tasks, the results among the three implementations of TACTP have no significant difference in duration prediction task.

4.4 Analysis on temporal encoding

In this subsection, we will discuss the different approaches for handling time factor vectors in our proposed TACTP framework.

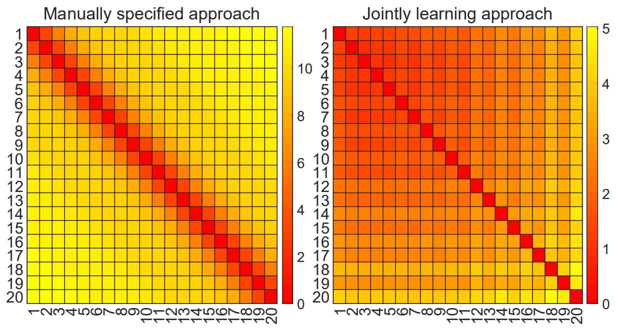


Figure 5: Analysis on temporal encoding. The two figures present the Euclidean distances among the latent time vectors obtained by manually specified and jointly learning approaches, respectively.

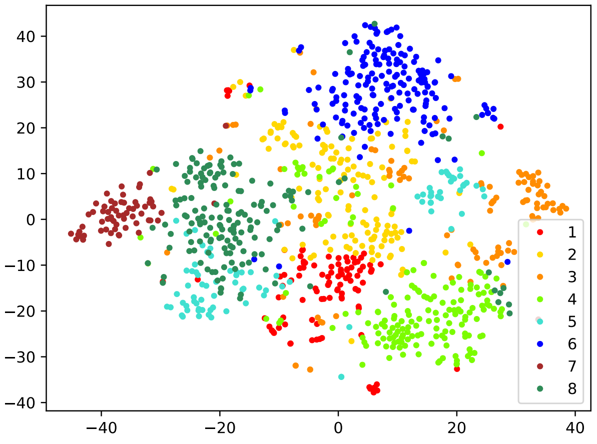


Figure 6: The visualization of TACTP based company clustering. (The clusters are distinguished by different colors.)

Specially, we compare the following variants and show the performance in Table 4: 1) All the three TACTP (M) variants adopts manually specified approach for temporal encoding while the three TACTP (J) variants adopt jointly learning approach; 2) TACTP-C (M) and TACTP-C (J) directly concatenate the talent representation with the working duration vector and seniority vector without time perception layer in both modeling and prediction stages; 3) TACTP-O (M) and TACTP-O (J) only exploit temporal information in modeling process but not prediction process, which means we directly use the latent talent vector u_{it} for prediction instead of time-aware vector $\widetilde{u}_{it}$; 4) TACTP-N ignores the temporal encoding mechanism and simply input the primary talent representations in both modeling and prediction stages.

First we can observe that TACTP-N performs worse than all the other variants, which clearly demonstrates the necessity of the temporal encoding mechanism. Moreover, TACTP-O (M) and TACTP-O (J) perform worse than TACTP (M) and TACTP (J), respectively. This demonstrates the importance of producing time-aware prediction results, but not just exploiting dynamic temporal information in the modeling stage. By comparing TACTP-C (M) and TACTP-C

Table 5: Analysis on position vectors obtained by TACTP. The values in the brackets are the cosine similarities between the latent vectors of the given position and the target position.

Target position	Positions and cosine similarities		
entrepreneurship	consulting (0.2432)	finance (0.2288)	operation (0.2009)
	community and social services (-0.1894)	quality assurance (-0.1797)	purchasing (-0.1686)
sale	market (0.2403)	accounting (0.2182)	operation (0.2102)
	quality assurance (-0.2647)	legal (-0.2531)	research (-0.1765)
program and project management	consulting (0.5006)	engineering (0.3857)	business development (0.3695)
	health-care services (-0.3719)	real estate (-0.3551)	legal (-0.2998)

(J) with TACTP (M) and TACTP (J), we can easily observe the large performance boost by our designed time perception layer, which shows that the time perception layer can successfully capture the individually customized temporal information. Lastly, variants with jointly learning approach significantly outperform manually specified approach by large margins, implying that jointly learning approach could learn better representations for fitting the complex temporal information in real-world data.

To further show the differences between manually specified and jointly learning approaches, we present the heatmap of the Euclidean distances among the latent time vectors obtained by these two approaches. A unit of abscissa or ordinate in Figure 5 means a half-year. Here we only show the vectors representing no more than 10 years, since the great majority of working durations are less than 10 years in our dataset. We can easily find that the Euclidean distance between two time values in jointly learning approach changes more smoothly than manually specified approach. Interestingly, we can observe the sawtooth tendency for the Euclidean distances among the latent time vectors, that is to say, the latent time vectors in odd half-years tend to have smaller distances with other time vectors. This phenomenon is exactly in accord with Figure 2(d), which shows that talents change their jobs more frequently in the first half of a year than the second half of a year. Our jointly learning approach is capable of learning this tendency automatically from the real-world data while manually specified approach has no such property. In summary, jointly learning approach can produce more proper latent vectors and better prediction performance than manually specified approach.

4.5 Analysis on latent position vectors

Our TACTP framework is able to learn a unique latent vector for each position. The larger similarity between two latent position vectors would imply more chances for talents to change their jobs between these two positions. Here we provide some cases to show the practical guiding significance of TACTP. Table 5 shows the top 3 positively related and negatively related positions to the target position. For example, we can observe from Table 5 that the positions with top 3 cosine similarities to "entrepreneurship" are "consulting", "finance", and "operation", while the positions with top 3 similarities from the bottom are "community and social services", "quality assurance", and "purchasing". Thus, if a talent wants to become an entrepreneur, she may first accumulate experiences in the related positions. Also, we can find that the most related position to "sale" is "market", since it is easy for talents to change their jobs between these two positions. Meanwhile, talents in "research", "legal", and "quality assurance" are not likely to transfer to "sale".

4.6 Analysis on latent company vectors

In this subsection, we provide an overall view of the companies in our dataset to show the interpretability of TACTP. The smaller distance between two latent company vectors indicate talent may be more likely to change their jobs between these two companies. We first performed k-means clustering [3] to partition all the 1, 002 companies into 8 clusters according to their latent company vectors in the first half of 2017 obtained by TACTP. Then we utilize the t-SNE algorithm [25] to transform the original 150-dimensional vector into a 2-dimensional space for visualization, as shown in Figure 6. We can observe that *cluster* 3 and 4 are quite close. Actually, companies in these two clusters are all high-tech companies, such as *Google*, *Apple*, *Microsoft*, *Facebook*, and *IBM*. Moreover, *cluster* 5 and 8 are also very close to and interlocked with each other in the low-dimensional space. In fact, the companies in these two clusters are both relevant to energy and manufacturing industries. The difference is that clusters 5 contains more military and aviation companies, such as *Lockheed Martin* and *Airbus*, while clusters 8 contains more home appliance companies, such as *Sony* and *Siemens*. Besides, *cluster* 6, which is the biggest cluster, is composed of many retail and hospitality industries, such as *Amazon*, *Best Buy*, *Hilton*, *7-Eleven*, *McDonald's*, and *KFC*. Differently, *cluster* 7, as the smallest and most concentrated cluster, mainly consists of health-care, pharmaceutical, and biotechnology industries, such as *Pfizer* and *Johnson & Johnson*. Lastly, in *cluster* 1 and *cluster* 2, the vast majority of companies belongs to banks, financial and insurance companies, such as *JPMorgan Chase & Co*, *BNP Paribas*, *Goldman Sachs*, and *AXA*. To sum up, our TACTP framework is capable of capturing the attributes of different companies automatically, and the results can be used for guiding talents to find out the suitable career transitions.

5 RELATED WORK

The related work can be classified into three main categories, i.e., career trajectory prediction, time sequential modeling, and collaborative filtering.

Career trajectory prediction. Career trajectory prediction is an important topic in human resource management [52, 53]. Traditional studies usually focused on the qualitative analysis [7, 41]. In recent years, there is an increasing interest in applying machine learning solutions, especially deep learning methods, to career trajectory modeling. Many of them focused on the transitions of companies and positions. For example, Li et al. [19] designed a recurrent neural network to predict the employee's next career move. Liu et al. [24] utilized a multi-task learning model for predicting career paths while He et al. [10] chose to predict job seekers' future job information by convolutional neural networks. Xu et al. [49] developed a deep sequence career trajectory prediction model based on the recurrent neural network model to predict whether there is a job change in six months. Recently, Meng et al. [27] predicted the next potential company and how long the talent will stay in the next job with an attention-based Long Short-Term Memory model. Besides the above works, there are also some researchers who tried to construct job recommender systems based on the prediction of career trajectories [22, 29, 38, 45, 51]. For example, Shalaby et al. [38] used a graph-based approach for building an item-based job recommender system. Besides, some works took the employees' skills into considerations for better job recommendation [6, 30].

Different from the above works, we provide a unified solution for all three job recommendation tasks (i.e., the next company, position, and current working duration) by integrating sequential modeling and collaborative filtering methods.

Time sequential modeling. Time sequential modeling has achieved great success in a variety of applications, such as recommender system, events prediction, time-evolving graphs, and so on [13, 17, 21, 32]. Early sequential prediction tasks focused on the sequential pattern mining [39] and transition modeling [36]. Since recurrent neural networks (RNNs) [12] (e.g., the well-known Long Short-Term Memory (LSTM) [9] and Gated Recurrent Unit (GRU) [5]), have achieved great success on various sequential modeling tasks due to their superior performance, they are also widely applied in personalized prediction systems. One representative application is sequential recommendation, which is based on the sequential prediction task. For instance, Wang et al. [46] modeled complicated interactions among multiple factors by using different aggregation operations over the representations. Quadrana et al. [33] designed a hierarchical recurrent neural network with cross-session information transfer. Li et al. [21] incorporated users' historical preferences and consumption motivations for next-item recommendation scenario.

While many current studies focus on sequences with fixed intervals [18], variable interval time sequences is still a great challenge, where the time intervals are different, and thus temporal information would be more influential in exploring the state changes at different time points and making next predictions. Quite recently, some works tried to capture the dynamic temporal information in the sequences. For example, Pavlovski et al. [31] calculated a temporal score to measure the influence of irregular time intervals. Tan et al. [40] designed a dual-attention GRU to handle the missing values in time intervals for patients. Li et al. [18] chose to incorporate the time relation matrix into self-attention units. However, all of the above works just consider the dynamic temporal information in the modeling process but not the prediction process, and thus cannot handle the varying user status for the next prediction. In this paper, we design a general temporal encoding mechanism for both the modeling and prediction stages.

Collaborative filtering. Generally, collaborative filtering (CF) methods can be further divided into two classes, i.e., neighborhood based and latent factor based methods. Neighborhood based CF methods [37] usually first search the nearest neighbors from the user or item aspect and then make recommendations according to the neighbors' records. By comparison, latent factor based CF methods choose to project users and items into latent factor space. For example, probabilistic matrix factorization [28], as one of the most widely used latent factor models, factorized the rating matrix into the product of user and item latent vectors in a low-rank space. Recently, many researchers began to combine neural networks with CF methods. For instance, He et al. [11] leveraged a multi-layer perceptron to learn the user-item interaction function and Xue et al. [50] designed an implementation of matrix factorization by using multi-layer perception networks. Moreover, Wang et al. [44] and Li and She [20] exploited the stacked denoising autoencoders and variational autoencoders for combining collaborative filtering with deep content embeddings.

Some researchers tried to combine neighborhood based CF methods with job recommender systems [23, 51]. However, neighborhood based CF methods usually perform not well in sparse situations [15]. In this paper, we incorporate the latent factor based CF method in our proposed framework for generating high-quality personalized recommendations.

6 CONCLUSION

In this paper, we proposed a novel time-aware career trajectory prediction (TACTP) framework for jointly predicting the three key elements in career trajectory, i.e., timing, company, and position. A unique perspective of TACTP is that we can generate time-aware predictions according to the varying duration of the current job owing to our proposed temporal encoding mechanism. Specifically, we first developed a unified time-aware sequential model based on recurrent networks to map the heterogeneous inputs into latent factor vectors from time, company, and position perspectives for each talent. Then we combined the talent representations with company and position representations to make predictions by latent factor based collaborative filtering. Finally, we conducted extensive experiments on a large-scale real-world dataset to demonstrate the effectiveness of TACTP.

ACKNOWLEDGMENTS

This research was supported by grants from the National Natural Science Foundation of China (No.61836013, 91746301).

REFERENCES

[1] 2017. https://www.forbes.com/sites/joshbersin/2017/05/26/google-for-jobs-potential-to-disrupt-the-200-billion-recruiting-industry/.
[2] William J Anderson. 2012. *Continuous-time Markov chains: An applications-oriented approach.* Springer Science & Business Media.
[3] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.
[4] Wenbo Chen, Pan Zhou, Shaokang Dong, Shimin Gong, Menglan Hu, Kehao Wang, and Dapeng Wu. 2018. Tree-Based Contextual Learning for Online Job or Candidate Recommendation With Big Data Support in Professional Social Networks. *IEEE Access* 6 (2018), 77725–77739.
[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
[6] Vachik S Dave, Baichuan Zhang, Mohammad Al Hasan, Khalifeh AlJadda, and Mohammed Korayem. 2018. A combined representation learning approach for

better job and skill recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1997–2005.

[7] Thomas H Davenport, Jeanne Harris, and Jeremy Shapiro. 2010. Competing on talent analytics. *Harvard business review* 88, 10 (2010), 52–58.

[8] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1243–1252.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.

[10] Miao He, Dayong Shen, Yuanyuan Zhu, Renjie He, Tao Wang, and Zhongshan Zhang. 2019. Career Trajectory Prediction based on CNN. In *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE, 22–26.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 173–182.

[12] John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79, 8 (1982), 2554–2558.

[13] Binbin Jin, Hongke Zhao, Enhong Chen, Qi Liu, and Yong Ge. 2019. Estimating the days to success of campaigns in crowdfunding: A deep survival perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4023–4030.

[14] Alan Karr. 2017. *Point processes and their statistical inference*. Routledge.

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[16] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[17] Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. 2019. Predicting Path Failure In Time-Evolving Graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1279–1289.

[18] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 322–330.

[19] Liangyue Li, How Jing, Hanghang Tong, Jaewon Yang, Qi He, and Bee-Chung Chen. 2017. Nemo: Next career move prediction with contextual embedding. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 505–513.

[20] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 305–314.

[21] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1734–1743.

[22] Kuan Liu, Xing Shi, Anoop Kumar, Linhong Zhu, and Prem Natarajan. 2016. Temporal learning and sequence modeling for a job recommender system. In *Proceedings of the Recommender Systems Challenge*. 1–4.

[23] Rui Liu, Wenge Rong, Yuanxin Ouyang, and Zhang Xiong. 2017. A hierarchical similarity based job recommendation service framework for university students. *Frontiers of Computer Science* 11, 5 (2017), 912–922.

[24] Ye Liu, Luming Zhang, Liqiang Nie, Yan Yan, and David S Rosenblum. 2016. Fortune teller: predicting your career path. In *Thirtieth AAAI conference on artificial intelligence*.

[25] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[26] Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*. 6754–6764.

[27] Qingxin Meng, Hengshu Zhu, Keli Xiao, Le Zhang, and Hui Xiong. 2019. A Hierarchical Career-Path-Aware Neural Network for Job Mobility Prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 14–24.

[28] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[29] Ioannis Paparrizos, B Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems*. 325–328.

[30] Bharat Patel, Varun Kakuste, and Magdalini Eirinaki. 2017. CaPaR: a career path recommendation framework. In *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 23–30.

[31] Martin Pavlovski, Jelena Gligorijevic, Ivan Stojkovic, Shubham Agrawal, Shabhareesh Komirishetty, Djordje Gligorijevic, Narayan Bhamidipati, and Zoran Obradovic. 2020. Time-Aware User Embeddings as a Service. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3194–3202.

[32] Mathias Perslev, Michael Jensen, Sune Darkner, Poul Jørgen Jennum, and Christian Igel. 2019. U-Time: A Fully Convolutional Network for Time Series Segmentation Applied to Sleep Staging. In *Advances in Neural Information Processing Systems*. 4417–4428.

[33] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 130–137.

[34] Narda R Quigley and Walter G Tymon. 2006. Toward an integrated model of intrinsic motivation and career self-management. *Career development international* (2006).

[35] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

[36] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[37] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[38] Walid Shalaby, BahaaEddin AlAila, Mohammed Korayem, Layla Pournajaf, Khalifeh AlJadda, Shannon Quinn, and Wlodek Zadrozny. 2017. Help me find a job: A graph-based approach for job recommendation at scale. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 1544–1553.

[39] Shuo Shang, Ruogu Ding, Kai Zheng, Christian S Jensen, Panos Kalnis, and Xiaofang Zhou. 2014. Personalized trajectory matching in spatial networks. *The VLDB Journal* 23, 3 (2014), 449–468.

[40] Qingxiong Tan, Mang Ye, Baoyao Yang, Siqi Liu, Andy Jinhua Ma, Terry Cheuk-Fung Yip, Grace Lai-Hung Wong, and PongChi Yuen. 2020. DATA-GRU: Dual-Attention Time-Aware Gated Recurrent Unit for Irregular Multivariate Time Series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 930–937.

[41] Ibraiz Tarique and Randall S Schuler. 2010. Global talent management: Literature review, integrative framework, and suggestions for further research. *Journal of world business* 45, 2 (2010), 122–133.

[42] Mingfei Teng, Hengshu Zhu, Chuanren Liu, Chen Zhu, and Hui Xiong. 2019. Exploiting the Contagious Effect for Employee Turnover Prediction. (2019).

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[44] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[45] Pengyang Wang, Yingtong Dou, and Yang Xin. 2016. The analysis and design of the job recommendation model based on GBRT and time factors. In *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*. IEEE, 29–35.

[46] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. 403–412.

[47] Francisco Wilhelm and Andreas Hirschi. 2019. Career self-management as a key factor for career wellbeing. In *Theory, Research and Dynamics of Career Wellbeing*. Springer, 117–137.

[48] Huang Xu, Zhiwen Yu, Hui Xiong, Bin Guo, and Hengshu Zhu. 2015. Learning career mobility and human activity patterns for job change analysis. In *2015 IEEE International Conference on Data Mining*. IEEE, 1057–1062.

[49] Huang Xu, Zhiwen Yu, Jingyuan Yang, Hui Xiong, and Hengshu Zhu. 2018. Dynamic talent flow analysis with deep sequence prediction modeling. *IEEE Transactions on Knowledge and Data Engineering* 31, 10 (2018), 1926–1939.

[50] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*. 3203–3209.

[51] Shuo Yang, Mohammed Korayem, Khalifeh AlJadda, Trey Grainger, and Sriraam Natarajan. 2017. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowledge-Based Systems* 136 (2017), 37–45.

[52] Denghui Zhang, Junming Liu, Hengshu Zhu, Yanchi Liu, Lichen Wang, Pengyang Wang, and Hui Xiong. 2019. Job2Vec: Job title benchmarking with collective multi-view representation learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2763–2771.

[53] Le Zhang, Hengshu Zhu, Tong Xu, Chen Zhu, Chuan Qin, Hui Xiong, and Enhong Chen. 2019. Large-Scale Talent Flow Forecast with Dynamic Latent Factor Model?. In *The World Wide Web Conference*. 2312–2322.