Attentive Heterogeneous Graph Embedding for Job Mobility Prediction

Le Zhang[1], Ding Zhou[1], Hengshu Zhu[2]*, Tong Xu[1]*, Rui Zha[1], Enhong Chen[1], Hui Xiong[3]*
[1] School of Computer Science and Technology, University of Science and Technology of China,
[2]Baidu Talent Intelligence Center, Baidu Inc, [3]Rutgers University
{zhangle0202, zhoudinglive}@gmail.com, zhuhengshu@baidu.com, zr990210@mail.ustc.edu.cn,
{tongxu, cheneh}@ustc.edu.cn, hxiong@rutgers.edu

ABSTRACT

Job mobility prediction is an emerging research topic that can benefit both organizations and talents in various ways, such as job recommendation, talent recruitment, and career planning. Nevertheless, most existing studies only focus on modeling the individual-level career trajectories of talents, while the impact of macro-level job transition relationships (e.g., talent flow among companies and job positions) has been largely neglected. To this end, in this paper we propose an enhanced approach to job mobility prediction based on a heterogeneous company-position network constructed from the massive career trajectory data. Specifically, we design an **A**ttentive **h**eterogeneous graph **e**mbedding for sequenti**a**l pre**d**iction (Ahead) framework to predict the next career move of talents, which contains two components, namely an attentive heterogeneous graph embedding (AHGN) model and a Dual-GRU model for career path mining. In particular, the AHGN model is used to learn the comprehensive representation for company and position on the heterogeneous network, in which two kinds of aggregators are employed to aggregate the information from external and internal neighbors for a node. Afterwards, a novel type-attention mechanism is designed to automatically fuse the information of the two aggregators for updating node representations. Moreover, the Dual-GRU model is devised to model the parallel sequences that appear in pair, which can be used to capture the sequential interactive information between companies and positions. Finally, we conduct extensive experiments on a real-world dataset for evaluating our Ahead framework. The experimental results clearly validate the effectiveness of our approach compared with the state-of-the-art baselines in terms of job mobility prediction.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

Job Mobility Prediction; Graph Embedding; Sequential Modeling

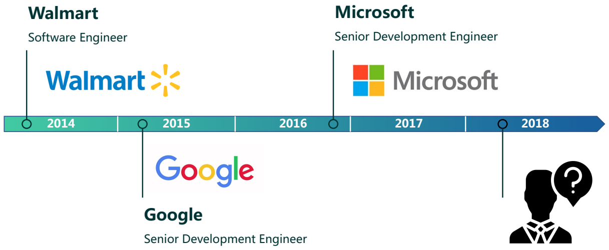


Figure 1: An example of a member's career trajectory.

ACM Reference Format:
Le Zhang[1], Ding Zhou[1], Hengshu Zhu[2]*, Tong Xu[1]*, Rui Zha[1], Enhong Chen[1], Hui Xiong[3]*. 2021. Attentive Heterogeneous Graph Embedding for Job Mobility Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467388

1 INTRODUCTION

The phenomenon of job hopping has become a new normal in the talent-economy era. Therefore, the research on job mobility prediction emerges as the times require, which can benefit both organizations and talents in various ways, such as competitive analysis, job recommendation, talent recruitment, and career planning. Traditional studies on job mobility prediction mainly focus on the determining factors [31, 33] and assessment [13, 36] of job mobility, based on social surveys or interviews. For example, Ng *et al.* [31] explored some intrinsic and extrinsic factors of job mobility, such as economic conditions, industry differences, personality traits, and desirability of mobility. Shockley *et al.* [36] created and validated a measure of subjective career success for individuals.

Recently, the rapid prevalence of online professional networks (OPNs) has enabled the accumulation of massive digital resumes, which opens an unparalleled opportunity for developing the data-driven intelligent approach to job mobility prediction [22, 30, 41]. For example, Li *et al.* [22] proposed an encoder-decoder framework to integrate the individual profiles to predict the next career move of talents. Meng *et al.* [30] proposed a hierarchical neural network to integrate three levels of individual information for job mobility prediction. Nevertheless, most existing studies only focus on modeling the individual-level career trajectories of talents, while the impact of macro-level job transition relationships (e.g., talent flow among companies and job positions) has been largely neglected. Indeed, it is intuitive that the macro-level job transition information may reflect the competitiveness and trend of talent market, which

*Corresponding authors.

KDD '21, August 14–18, 2021, Virtual Event, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08...$15.00
https://doi.org/10.1145/3447548.3467388

will consequently influence the job hopping decision of individuals. Meanwhile, existing studies usually tend to represent the entities (e.g., companies and positions) based on the predefined attributes, which may suffer from the insufficient data and cannot model the entities comprehensively.

Therefore, in this paper, we propose to study the problem of job mobility prediction by exploring the impact of macro-level job transition relationships. Specifically, we design an **A**ttentive **h**eterogeneous graph **e**mbedding for sequenti**a**l pre**d**iction (Ahead) framework for enhancing job mobility prediction. In general, two major challenges will be addressed in the framework. First, a comprehensive representation of company and position should be generated with the consideration of their global and multiple relationships. Second, the mutual dependency between company and position should be carefully integrated. To this end, we first construct a heterogeneous company-position network by mining the massive career trajectory data, where the nodes represent all companies and job positions, the edges contain the different relationships of nodes (i.e., the job transitions between two companies or positions, and the belonging relationship of company and position). Then, we construct the first component of Ahead, namely attentive heterogeneous graph embedding (AHGN) model, to represent companies and positions comprehensively based on the graph neural network. In particular, to distinguish the heterogeneity of nodes, two aggregators are designed to integrate neighbor information. The external aggregator is used to aggregate the information of neighbors with different types according to the graph convolutional rule. The internal aggregator is employed to aggregate the information of neighbors with same type, in which a transition-aware attention is used to integrate the contextual features of nodes. Afterwards, a novel type-attention mechanism is proposed to automatically learn the importance of internal and external aggregators for updating nodes representation. The other component of the Ahead framework is career path mining, where we first describe the career trajectory as two sequences of company and position, and then design a novel Dual-GRU model to capture the sequential interactive information between company and position by integrating the hidden states of two sequences with an attention mechanism. Finally, the outputs of the Dual-GRU model are used to predict job mobility by the fully-connected layers. Specifically, the major contributions of this paper can be summarized as follows:

- We propose to study the problem of job mobility prediction by exploring the impact of macro-level job transition relationships, which fills the research void in previous studies that only model the individual-level career trajectories.
- We design a novel attentive heterogeneous graph embedding framework for enhancing job mobility prediction, where an AHGN model is used to learn the comprehensively representations of company and position, and a novel Dual-GRU model is applied to model the career path with the consideration of the mutual influence between company and position.
- We conduct extensive experiments on a real-world dataset for evaluating our Ahead framework, and the experimental results clearly validate the effectiveness of our approach compared with the state-of-the-art baselines in terms of job mobility prediction.

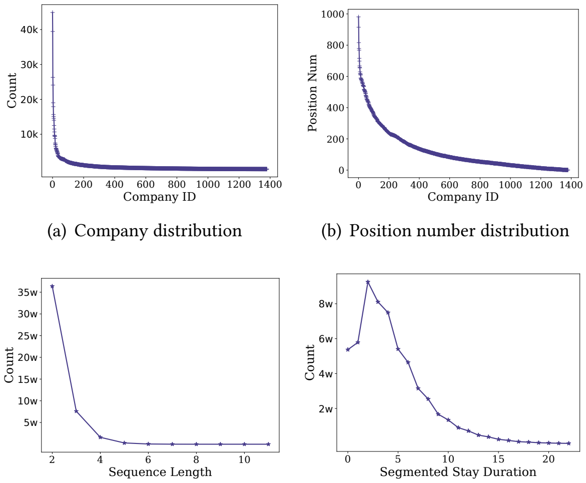
2 RELATED WORK

The related work can be summarized into three main categories, namely *job mobility analysis*, *sequence forecasting* and *network representation learning*.

Job Mobility Analysis. Job mobility analysis is a hot topic in human resource management. Traditional studies mainly focus on the determining factors and assessment of job mobility. For example, Pan et al. [33] analyzed how factors such as personality, industry and education background impact career paths, Ng et al. [31] introduced a multi-level theoretical framework to describe how individual job mobility unfolds, and Shockley *et al.* [36] created and validated an index of subjective career success for individuals. Recently, data mining techniques have been widely applied to the job mobility analysis tasks, including individual turnover prediction [21, 38], career trajectory modeling [42], job mobility prediction [22, 30, 41], competitive analysis [44] and so on. In this paper, we mainly focus on the issue of job mobility prediction, in which the prediction targets include employers, positions, working duration and so on. For instance, Li et al. [22] proposed a contextual LSTM model to integrate the profile context and career path dynamics simultaneously for predicting the next company/position of talents. Meng et al. [30] proposed a hierarchical career-path-aware neural network to model the individual job mobility, which predicted the next employer and the corresponding working duration for talents. In general, most existing studies focus on modeling the individual-level career trajectories of talents. Differently, we propose to study the problem of job mobility by exploring the impact of macro-level job transition relationships in the view of heterogeneous company-position network.

Sequence Forecasting. Regarding the sequence forecasting problem, several modeling methods have been proposed. For instance, the CTMC model [1] uses the stochastic probability to describe a series of events, in which the state space is discrete but has continuous time. At the same time, the CRF model [19] allows long-distance dependencies, and integrates rich features for sequence forecasting. Correspondingly, we design our solution based on the Recurrent Neural Networks (RNN) [16], which has achieved state-of-art performance on sequential modeling tasks, including speech recognition [9, 10], machine translation [6, 7] and recommendation [26, 28]. However, the RNN model suffers from the vanishing gradient problems [14]. To address this issue, the variations of RNN, such as LSTM [15] and GRU [6] are proposed by introducing several gates in neural cells to gain better long-term memory efficiency. Apart from that, the attention mechanism is further introduced to improve the prediction performance of RNNs [2, 29]. Different from the existing methods, in this paper, we propose a new RNN structure, namely Dual-GRU, to model the parallel sequences of company and position, and capture their mutual influence with attention mechanism.

Network Representation Learning. Representation Learning aims to automatically discover the representations needed for the downstream applications, which is common in smart services [23, 25, 47]. As a branch, network representation learning is proposed to embed node into a low dimensional space while preserving the network structure and property. Large efforts have been made on this issue, such as the matrix factorization based models [3, 32],



(a) Company distribution (b) Position number distribution

(c) Sequence length distribution (d) Duration distribution

Figure 2: The data distribution of different aspects.

and the random walk based models [11, 35]. Recently, the graph neural networks (GNNs) are proposed to represent nodes by using the rich neighborhood information. For example, GCN [18], GraphSAGE [12] and GAT [37] employ convolutional operator, LSTM architecture, and self-attention mechanism to aggregate the feature information of neighboring nodes respectively. However, all these algorithms are proposed for the homogeneous graph. As a result, several network embedding solutions have been expanded to heterogeneous graphs [5, 39]. For instance, the heterogeneous skip-gram model based methods [8, 17] conduct the meta-path based random walks to generate graph contexts. Beyond the random walk based model, HAHE [46], HGAT [24] and HAN [40] apply different attention architectures to integrate the different type features of different neighborhoods. For example, HAHE [46] employs a hierarchical attentive structure to capture the personalized preferences on meta paths and path instances in each semantic space. Different from these models mentioned above, in this paper, we propose a new heterogeneous graph embedding structure which is well-designed for the heterogeneous company-position graph.

3 PRELIMINARIES

In this section, we will first introduce the real-world dataset in our paper. Then, several pre-studies on the dataset will be introduced. At last, we formally define the problem of job mobility prediction.

3.1 Data Description

The data set were collected from one of the largest online professional social platforms, i.e., Linkedin, where users can create professional resumes to share their working and education experience. In detail, we extracted the individual profiles as well as the working records from these resumes. Specifically, each profile consists of a user name and self-description, and each working record is composed of company name, job position and working duration. In addition to the individual information, we also collected some static features of the companies from Linkedin, including company type, size, etc. We will introduce the processing details in section A.1 of the Appendix. Through the data pre-processing, we can extract the career trajectory of each talent, as shown in Figure 1. Furthermore, we analyzed the distribution of career trajectory records from different aspects, as shown in Figure 2. Obviously, the distribution of data is imbalanced, and we need to deal with the imbalanced distribution for better predictions.

3.2 Data Exploration

Next, we conduct pre-studies to analyze several determining factors that affect job mobility from the macro view.

Firstly, we will explore the relationship between company similarity and job transition. For each company, we collected all positions that company contains. Afterwards, we used the labeled dataset [27] to extract the *function* words from all positions, which can describe the business function of companies. For example, for the position "software engineer", "software" is a *function* word, which indicates the company is related to IT. Then, we constructed a vector to represent each company, where the dimension was equal to the size of *function* words, and each dimension denotes the normalized frequency of word. The similarity between two companies can be defined as the dot product of their vectors. Finally, we used Pearson Correlation Coefficient (PCC) metric [34] to measure the correlation between company similarity and job transition. The PCC score is 0.6376 with P-value nearly 0, which indicates the company similarity and job transition are strongly correlated.

Secondly, we discuss the relationships between position similarity and job transition. As mentioned in [43], words of job positions contain rich semantic information. Usually, if two positions have more identical words, they could be more similar to each other. For example, the position "software engineer" is more similar to "software developer" than "account manager". Thus, we split the job titles by word and analyzed the number of changed words during job transitions. The statistics results show that more than 60% of the job transitions are generated between two job positions which at least exists one identical word. Obviously, job transitions usually occur more between two similar positions.

Thirdly, we analyze the relationships between companies and positions. We first extracted the correspondence between companies and positions. For each position, we maintained a list of companies that this position belongs to, and vice versa. According to these two lists, we find that when a job transition occurs within the job mobility trajectory records, the destination company is probably within the list of current position, with the probability higher than 85%. Similarly, for the new position, it is also probably within the list of current company, with a probability higher than 85%. This situation indicates that both current company and position could benefit the prediction task of future mobility selection.

In summary, these pre-studies motivate us to construct a graph structure to capture the global relationships among companies and positions for better job mobility prediction.

3.3 Problem Formulation

Here we first define the personal career trajectory and the heterogeneous company-position network, and then formally formulate the job mobility prediction problem.

Definition 1 (Career Trajectory). *The career trajectory of a person u is an ordered sequence of jobs, which can be summarized as $\mathcal{J}(u) = \{\mathcal{J}_1, \mathcal{J}_2, ..., \mathcal{J}_L|u\}$, where $\mathcal{J}_i$ is the i-th job record of u, denoted by a tuple, i.e., $\mathcal{J}_i = (c_i, p_i, d_i)$, indicating that user u worked at company c_i with position p_i and the stay time is d_i.*

It is noticed that the working duration in the last record, i.e., d_L, is unknown, because we usually do not know how long the talents will stay in current company until they move to the next company and update their resume. According to the career trajectory, we can extract two sequences for company and position respectively. Specifically, the company sequence can be written as:

$$\mathcal{S}_c(u) = \{(c_1, c_2, ..., c_L)|u\}, \tag{1}$$

and the position sequence can be written as:

$$\mathcal{S}_p(u) = \{(p_1, p_2, ..., p_L)|u\}. \tag{2}$$

The two adjacent companies and positions in a sequence can construct an edge, e.g., $<c_i, c_{i+1}>$ and $<p_i, p_{i+1}>$, which represents the connectivity between companies/positions. And the strength of the connection is determined in a heuristic way by the frequency of a pair in all trajectories. Moreover, in each step i, we have a company-position pair, e.g., $<c_i, p_i>$, which represents the ownership relationship between company and position. As a result, by treating each company or position as a node and link them by the corresponding relationships, we can construct a heterogeneous company-position network, which is defined as follows:

Definition 2 (Company-Position Network). *The company-position network is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = (\mathcal{V}_c \cup \mathcal{V}_p)$ is the set of nodes, $\mathcal{E} = (\mathcal{E}_{cc} \cup \mathcal{E}_{cp} \cup \mathcal{E}_{pp})$ is the set of edges, $\mathcal{V}_c$ presents the set of companies, and $\mathcal{V}_p$ presents the set of job positions. Specifically, each edge in $\mathcal{E}_{cc}$ indicates the job transitions between two companies, each edge in $\mathcal{E}_{pp}$ indicates the job transitions between two positions, and each edge in $\mathcal{E}_{cp}$ indicates whether the position belongs to the company.*

With the above definition, we formulate the job mobility prediction problem as follows:

Definition 3 (Job Mobility Prediction Problem). *Given a dataset $\mathcal{D}$ consisting of career trajectories of talents from set U, for a query $q : \{\mathcal{J}(u), \Omega(u)\}$ from talent $u \notin U$, where $\Omega(u)$ denotes the personal-specific features. Our target is to predict u's next career move, including company c_{L+1}, position p_{L+1} and duration d_L.*

4 TECHNICAL DETAIL

In this section, we will introduce the technical details of our Ahead framework. As shown in Figure 3, our framework mainly consists of three components, namely *Attentive Heterogeneous Graph Embedding* (AHGN) to learn the comprehensive representation for company and position, the *Career Path Mining* to model the individual sequential trajectory, and the *Prediction Module* to integrate the individual sequential information to predict job mobility.

4.1 Attentive Heterogeneous Graph Embedding

As we all know, graph neural network (GNN) have been widely studied in many scenarios and achieved major success on the general graph learning problem. Among these achievements, the core idea of message-passing neural networks (MPNNs) is to generate node embedding vector by aggregating the features of node's neighborhoods. Inspired by this, we extend the graph neural network to learn the heterogeneous graph embedding. We define the aggregation process for the different types of nodes as external aggregation, and the same type as internal aggregation. Along this line, we first use both external and internal aggregation modules to aggregate different types of information, and then design a type-level attention mechanism to fuse them for fully representing nodes.

4.1.1 ***External aggregation.*** The external aggregation is to aggregate the information from neighbors with different types. For example, in terms of a position, the external aggregation is used to aggregate information from its neighbors of company type. Therefore, in this part, we mainly focus on the sub-graph $\mathcal{G}_{cp} = \{\mathcal{V}, \mathcal{E}_{cp}\}$, which only contains the company-to-position edges.

At first, it is obvious that different types of nodes have different feature spaces. For example, the features of company include size, location, etc, while the features of position include function, responsibility, etc. Formally, let $\mathbf{z}_i$ denote the feature embedding of node i. To make the aggregation process feasible, we design the type-specific transformation matrix $\mathbf{W}_\tau$ to project the features of different types of nodes into the same feature space, and the projected feature of node i is defined as follows:

$$\hat{\mathbf{z}}_i = \mathbf{W}_\tau \cdot \mathbf{z}_i. \tag{3}$$

Indeed, the attributes of companies and positions are sparse, integrating the information from heterogeneous neighbors can alleviate this issue. Considering the belonging relationship between company and position is unweighted and undirected, we adopt the graph convolutional rule based on symmetric normalized Laplacian to construct the external aggregator:

$$\mathbf{a}_i^{(l)} = \sum_{j \in \mathcal{N}_E(i)} \frac{1}{\sqrt{|\mathcal{N}_E(i)||\mathcal{N}_E(j)|}} \mathbf{W}_E^{(l)} \mathbf{H}_j^{(l)}, \tag{4}$$

where $\mathcal{N}_E(i)$ is the set of neighbors for node i with different types, and $\mathbf{H}_j^{(l)}$ is the hidden representation of node j in l-th layer. Initially, $\mathbf{H}_i^{(0)} = \hat{\mathbf{z}}_i$. $\mathbf{W}_E^{(l)}$ is a layer-specific trainable transformation matrix.

4.1.2 ***Internal aggregation.*** The internal aggregation is to aggregate the information from nodes of the same type. Hence, we focus on the sub-graphs $\mathcal{G}_{cc}$ and $\mathcal{G}_{pp}$ which represent the job transition network of company and position respectively. Job transition relationship has several important attributes [43], i.e., the total number and the average working duration. Intuitively, a node (i.e., company or position) has a larger impact on nodes with more job transitions compared with nodes with fewer job transitions. Besides, the shorter average duration of job transition between a pair of nodes (i.e., company-to-company or position-to-position) usually leads to more similar nodes.

However, the message-passing neural networks (MPNNs) cannot take advantage of these job transition attributes, because the aggregation of MPNNs treats all neighbors of node equally. To overcome this problem, we propose a transition-aware attention mechanism to aggregate node representation features:

$$\mathbf{b}_i^{(l)} = \sum_{j \in \mathcal{N}_I(i)} att(\mathbf{H}_i^{(l)}, \mathbf{H}_j^{(l)}, \mathbf{r}_{ij}) \mathbf{H}_j^{(l)}, \tag{5}$$

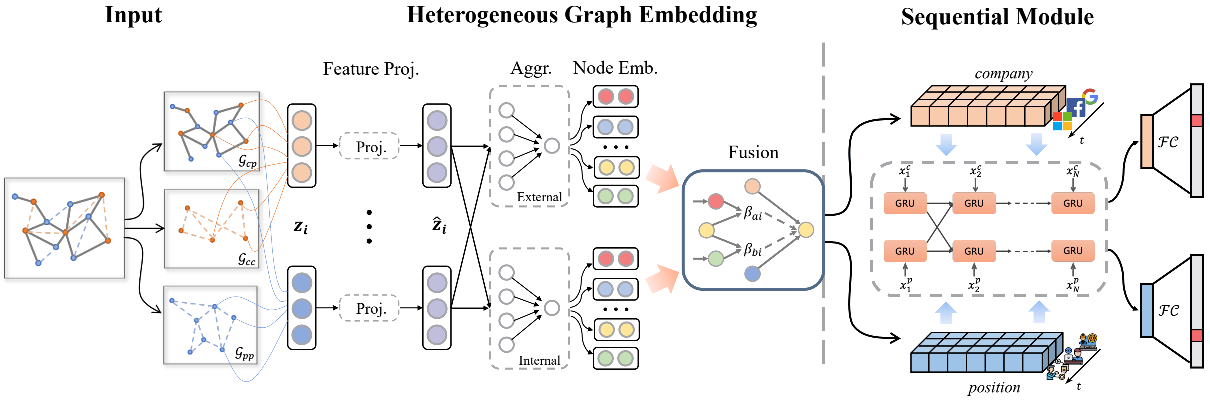


Figure 3: The diagrammatic sketch of the proposed Ahead framework on job mobility prediction.

where $\mathcal{N}_I(i)$ is the set of neighbors for node i with same type. $att(\mathbf{H}_i, \mathbf{H}_j, \mathbf{r}_{ij})$ is the transition-aware attentive weight of each neighbor j for i with their corresponding job transition feature $\mathbf{r}_{ij}$.

Both the total number and the average duration of job transition are real number. To embed the total number attribute, we first collect all the values, then discretize them into consecutive value bins evenly, where each bin can be regarded as a category. Afterwards, we randomly initialize the embedding for each category. Finally, the embedding would be jointly learned. Analogously, we conduct a similar operation on the average duration attribute.

Formally, let $\mathbf{n}_{ij}$ and $\mathbf{m}_{ij}$ represent the embedding of the number attribute and the duration attribute for the job transition from node i to node j. We concatenate the two embeddings and apply a dense layer transformation to represent the job transition:

$$\mathbf{r}_{ij} = \mathbf{W}_r \cdot (\mathbf{n}_{ij} \oplus \mathbf{m}_{ij}), \tag{6}$$

where $\mathbf{W}_r$ is transform matrix, $\oplus$ is concatenation operation.

Next, we implement the transition-aware attention mechanism by considering the features of nodes as well as the job transition features between nodes, which is formulated as follows:

$$att(\mathbf{H}_i, \mathbf{H}_j, \mathbf{r}_{ij}) = \frac{\exp(\boldsymbol{\mu}_1 \cdot (\mathbf{W}_1\mathbf{H}_i + \mathbf{W}_2\mathbf{H}_j + \mathbf{W}_3\mathbf{r}_{ij}))}{\sum_{k \in \mathcal{N}_I(i)} \exp(\boldsymbol{\mu}_2 \cdot (\mathbf{W}_1\mathbf{H}_i + \mathbf{W}_2\mathbf{H}_k + \mathbf{W}_3\mathbf{r}_{ik}))}, \tag{7}$$

where $\mathbf{W}_*$ are transform matrices, and $\boldsymbol{\mu}_*$ are the attention vectors. By integrating the transition-aware attention, we can obtain the internal aggregation $\mathbf{b}_i$ for each node based on Equation 5.

4.1.3 ***Representation Fusion.*** After the external and internal aggregation process, we turn to update the representation of each node. Generally, in terms of a node, different types of neighboring nodes may have different impacts on it. To distinguish the impacts, we propose a novel type-level attention mechanism to automatically learn the importance of different neighboring types for each node, i.e., internal and external. Formally, given a node i, the corresponding embedding of type τ is defined as the sum of the neighboring node features of node i with type τ:

$$\mathbf{e}_{\tau i}^{(l)} = \sum_{j \in \mathcal{N}_\tau(i)} \mathbf{H}_j^{(l)}. \tag{8}$$

To learn the importance of each type for node i, we first concatenate the type embedding with the target node embedding. Then, we measure the importance of the specific type τ as the similarity of the concatenated embedding with a type-specific attention vector μ_τ. The importance of type τ for node i is calculated as follows:

$$\beta_{\tau i}^{(l)} = \sigma(\boldsymbol{\mu}_\tau \cdot [\mathbf{e}_{\tau i}^{(l)} \oplus \mathbf{H}_i^{(l)}]). \tag{9}$$

By integrating the type-level attention, then the overall aggregation among different types can be calculated as follows:

$$\boldsymbol{\omega}_i^{(l)} = \beta_{ai}^{(l)} \mathbf{a}_i^{(l)} \oplus \beta_{bi}^{(l)} \mathbf{b}_i^{(l)}, \tag{10}$$

where $\beta_{ai}^{(l)}$ and $\beta_{bi}^{(l)}$ stand for the attention scores of external and internal type for node i respectively. $\mathbf{a}_i^{(l)}$ and $\mathbf{b}_i^{(l)}$ can be calculated by Equation 4 and 5 respectively. Afterwards, the representation of node i can be updated as follows:

$$\mathbf{H}_i^{(l+1)} = \sigma(\mathbf{W}_a \cdot \boldsymbol{\omega}_i^{(l)}), \tag{11}$$

where $\mathbf{W}_a$ is transform matrix, $\sigma(\cdot)$ is the activation function. The representation of node i in the last layer is treated as the final representation, denoted by $\mathbf{H}_i^*$.

4.2 Career Path Mining

After obtaining the representation of companies and positions, we turn to model the individual career paths.

As mentioned before, the career trajectory of a talent u can be described by two sequences, i.e., $\mathcal{S}_c(u) = (c_1, c_2, ..., c_L|u)$ and $\mathcal{S}_p(u) = (p_1, p_2, ..., p_L|u)$. To represent each company (or position) in sequence, two factors should be considered. The first one is time, obviously, the impact of a person's stay in the company or position for different periods of time is different. So we integrate the duration information to address this issue. Since the duration in the last record d_L is unknown, we construct a duration sequence $(d_0, d_1, ..., d_{L-1})$ of length L by adding $d_0 = 0$. By this way, the duration sequence can be aligned with company (or position) sequence. The second factor is personal information, as the same work experience may have different effects on different people. Let $\mathbf{H}_t^*$ denote the embedding of the t-th entity from AHGN, $\mathbf{E}_u$ represent the static individual features and $\mathbf{D}_t$ denote embedding of the t-th duration in the duration sequence. Then, the time-aware representation of company and position in u's trajectory can be defined as:

$$\mathbf{x}_t = \mathbf{W}_u \cdot (\mathbf{H}_t^* \oplus \mathbf{D}_t \oplus \mathbf{E}_u), \tag{12}$$

where $\mathbf{W}_u$ is transform matrix. We randomly initialize the embedding for all duration values, then the duration embedding will be updated during the training process. After that, the company and position sequences can be represented by $(\mathbf{x}_1^c, \mathbf{x}_2^c, ..., \mathbf{x}_L^c|u)$ and $(\mathbf{x}_1^p, \mathbf{x}_2^p, ..., \mathbf{x}_L^p|u)$.

In order to model the career sequential information, we take advantage of GRU as the basic model because it can alleviate the gradient vanishing problem in long-distance dependent sequential problems, as well as its efficiency compared with LSTM. In terms of GRU, the input is sequential vectors $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)$. At each time t, the input feature vector $\mathbf{x}_t$ is fed to a hidden cell which is identical for all time stamp, and the single cell is built as follows:

$$\begin{aligned}
r &= \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r),\\
z &= \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z),\\
\widetilde{\mathbf{h}}_t &= tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}(r \odot \mathbf{h}_{t-1}) + \mathbf{b}_h),\\
\mathbf{h}_t &= (1 - z) \odot \mathbf{h}_{t-1} + z \odot \widetilde{\mathbf{h}}_t.
\end{aligned} \tag{13}$$

In these equations, the parameters $\mathbf{W}_*$ denotes the weight matrices, $\mathbf{b}_*$ denotes bias, symbol $\odot$ denotes element-wise product operator, and $\mathbf{h}_t$ denotes the hidden state in time t.

Intuitively, we could construct two GRU for the company and position sequence respectively. However, modeling the two sequences separately may lose some important information since predictions for company and position are highly related. On the one hand, when predicting the next company, current position may limit the company selection range as proven in pre-study. For example, a software engineer of Google is more likely to choose an IT-related company. On the other hand, the job-hopping process involves job title benchmarking [43], which means the same job position in different companies may reflect different expertise levels. Thus current company is also a significant factor for position prediction.

To that end, we propose a Dual-GRU structure to model interactive information between companies and positions. It consists of two GRU that interact with each other for modeling company and position respectively. Taking the company sequence as an example, the basic GRU cell takes $\mathbf{x}_t^c$ and the predecessor hidden state $\mathbf{h}_{t-1}^c$ as inputs according to Equation 13. We enrich the inputs with the predecessor position hidden state $\mathbf{h}_{t-1}^p$, so that both the historical sequential information of company and position can be used for making predictions. To further distinguish the influence of two kinds of information, we apply the attention mechanism to automatically align weights for each part as:

$$\begin{aligned}
y_{t-1}^c &= \mathbf{W}(\mathbf{h}_{t-1}^c \oplus \mathbf{x}_t^c), \quad y_{t-1}^p = \mathbf{W}(\mathbf{h}_{t-1}^p \oplus \mathbf{x}_t^c),\\
\alpha_c &= \frac{\exp(y_{t-1}^c)}{\exp(y_{t-1}^c) + \exp(y_{t-1}^p)},\\
\mathbf{h}_{t-1}^{c*} &= \alpha_c \odot \mathbf{h}_{t-1}^c + (1 - \alpha_c) \odot \mathbf{h}_{t-1}^p,
\end{aligned} \tag{14}$$

where $\mathbf{W}$ denotes the transform matrix, and $\mathbf{h}_{t-1}^{c*}$ is the refined hidden state at $t-1$. Analogously, we can get $\mathbf{h}_{t-1}^{p*}$. With the new hidden state, the subsequent calculation is similar to the Equation 13.

4.3 The Prediction Module

Finally, we introduce the prediction module. Specifically, our career path prediction problem contains three major targets: the next company, the next position and the current working duration.

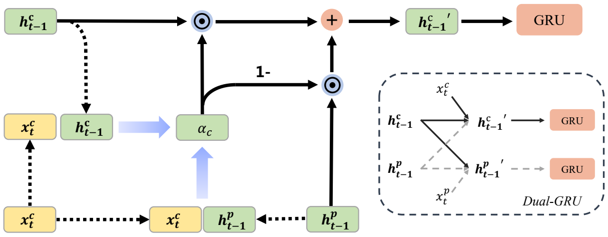


Figure 4: The diagrammatic sketch of Dual-GRU.

Intuitively, predicting the next company of talents can be formulated as a classification problem. Given a talent u, we firstly feed the current hidden state of company ($\mathbf{h}_t^c$) from Dual-GRU into a fully-connected layer, where the output dimension matches the total company number. Afterwards, we apply a softmax activation function to normalize the transition probability of each company, namely $\hat{\mathbf{o}}_t \in \mathbb{R}^{|\mathcal{V}_c|}$. Let $\mathbf{o}_t \in \mathbb{R}^{|\mathcal{V}_c|}$ denote the one-hot embedding of the t-th company in u's trajectory. Then, the loss function for next company prediction of talent u can be defined by the cross-entropy form:

$$\mathcal{L}_c^u = -\sum_{t=2}^{L} \mathbf{o}_t \log(\hat{\mathbf{o}}_{t+1}). \tag{15}$$

Analogously, the prediction process of next position is the same as predicting next company. We can obtain the loss function $\mathcal{L}_p^u$ for next company prediction of talent u.

Meanwhile, predicting the working duration of the current job can be formulated as a regression problem. Given a talent u, we concatenate the current hidden states of company and position ($\mathbf{h}_t^c$ and $\mathbf{h}_t^p$) from Dual-GRU. Then, we feed it into a fully-connected layer to transform it as the prediction $\hat{d}_t$, and the loss function for working duration prediction is defined as follows:

$$\mathcal{L}_d^u = \sum_{t=1}^{L-1} \frac{1}{2}(\hat{d}_t - d_t)^2. \tag{16}$$

Finally, the whole objective function is defined as follows:

$$\mathcal{L} = \sum_u (\mathcal{L}_c^u + \lambda_1 \mathcal{L}_p^u + \lambda_2 \mathcal{L}_d^u) + \lambda_3 ||\Theta||_2^2, \tag{17}$$

where λ_1 and λ_2 are hyper-parameters for balancing the different parts in the loss function. λ_3 is regularization parameter and $||\Theta||_2^2$ is the L2-norm over all parameters Θ.

5 EXPERIMENT

In this section, we will introduce the experimental details conducted on the real-world dataset for validating the proposed model.

5.1 Experiment Setup

5.1.1 Dataset. Our dataset contains 459,309 career trajectories, we constructed the heterogeneous company-position network from these trajectories. There are 1,380 companies and 2,098 positions, and the numbers of the three types of edges (i.e., $\mathcal{E}_{cc}$, $\mathcal{E}_{cp}$, $\mathcal{E}_{pp}$) are 90,000, 131,332, and 165,129 respectively.

5.1.2 Baselines. We compared Ahead with some state-of-art methods. Specifically, non-sequential models contain Logistic Regression (LR) and Random Forest (RF) [4]. Sequential models contain

Table 1: The overall performance of next company prediction, next position prediction and current duration prediction.

Methods	Company				Position				Duration	
	ACC@1	ACC@15	ACC@30	MRR	ACC@1	ACC@15	ACC@30	MRR	RMSE	MAE
LR	0.1316	0.4209	0.5136	0.1274	0.0702	0.2886	0.3837	0.0777	5.1272	4.3673
RF	0.2391	0.4428	0.5265	0.1641	0.1156	0.3759	0.4620	0.1143	4.9758	4.1707
LSTM	0.3636	0.6413	0.7081	0.4438	0.1685	0.5185	0.6191	0.2668	3.2390	2.4868
GRU	0.3868	0.6644	0.7288	0.4657	0.1775	0.5273	0.6270	0.2776	3.2984	2.5563
NEMO	0.4099	0.6863	0.7530	0.4888	0.2080	0.5829	0.6785	0.3151	3.1801	2.3949
HCPNN	0.4091	0.6691	0.7316	0.4839	0.2040	0.5848	0.6811	0.3120	3.2832	2.5387
HAN	0.4118	0.6608	0.7237	0.4835	0.1680	0.5191	0.6156	0.2671	3.1693	2.4114
HGAT	0.4127	0.6580	0.7216	0.4833	0.1722	0.5243	0.6200	0.2726	3.1684	2.4036
Ahead	**0.4267**	**0.6968**	**0.7622**	**0.5039**	**0.2083**	**0.5874**	**0.6820**	**0.3171**	**2.9176**	**2.0178**

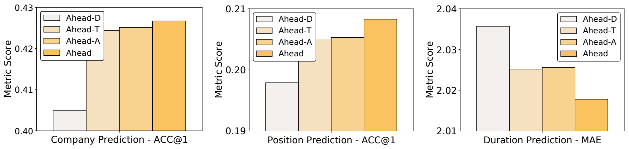


Figure 5: Ablation study on the job mobility prediction task.

LSTM [15], GRU [7], NEMO [22] and HCPNN [30]. NEMO and HCPNN are the most advanced models which are relevant to job mobility prediction. Finally, the heterogeneous graph embedding based models contain HAN [40] and HGAT [24], and we modified them to fit our problems. The detail of baselines will be introduced in section A.3 of appendix.

5.1.3 Evaluation Metrics. We used *Accuracy@k* (*Acc@k*) and *Mean Reciprocal Rank* (*MRR*) to evaluate the performance of next company and position prediction. And we selected *Root Mean Square Error* (*RMSE*) and *Mean Absolute Error* (*MAE*) for current duration prediction. The detail will be introduced in section A.2 of appendix.

5.2 Performance Evaluation

5.2.1 Overall Performance. The evaluation part includes prediction tasks for next company, next position and current duration. We randomly split all samples by (0.8/0.1/0.1) as the training/validation/test dataset respectively. Each method was trained on the training data, and the corresponding parameters were tuned on the validation data. The final performance was evaluated on the test data.

The results are summarized in Table 1. Obviously, our Ahead model achieves the best performance on all prediction tasks which clearly demonstrates the effectiveness of our model. Moreover, we have several observations. Firstly, the non-sequential models, i.e., LR and RF, always get the worst performance in all tasks, since they fail to handle the sequential information. Secondly, in terms of variants of RNN model, GRU gets better performance than LSTM, since our dataset is small and less frequent, which is more suitable for GRU learning. As the state-of-art models on job mobility prediction task, both NEMO and HCPNN have achieved competitive and stable performance, which indicates that integrating the individual information and designing effective strategies to model career path are quite useful. Finally, the modified state-of-art heterogeneous graph embedding methods, i.e., HGAT and HAN, can also get comparable performance, especially on the next company prediction task and working duration prediction task, which demonstrates that leveraging the heterogeneous graph can indeed improve the job mobility prediction performance. However, their performances drop a lot on the next position prediction task, which further indicates the robustness of the AHGN module of Ahead.

5.2.2 Ablation Study. To demonstrate the effectiveness of each component of Ahead, we conducted experiments on variants:

- **Ahead-D**: It replaces the Dual-GRU module with two independent GRU for company and position.
- **Ahead-A**: It drops the transition-aware attention mechanism in internal aggregator of AHGN.
- **Ahead-T**: There is no type-level attention in AHGN.

As shown in Figure 5, the performance of Ahead-D is significantly worse than other models, which indicates the effectiveness of the Dual-GRU module. Therefore, modeling the interaction between company and position can indeed improve prediction performance. By comparing Ahead with Ahead-A, it demonstrates the effectiveness of considering the job transition attributes between nodes. Moreover, by comparing Ahead with Ahead-T, it indicates that distinguishing the influence of external and internal neighbors can make better representation of nodes for better prediction.

5.2.3 Parameter Sensitivity. We also conducted two experiments to study how the input dimension of Dual-GRU and the category size of job transition features influence Ahead’s performance. Firstly, we discuss the sensitivity of the input dimension of Dual-GRU, which is summarized in Figure 6. In general, the performance is improved with increasing dimension size, since more dimensions may probably keep more useful information. Also, the performance keeps relatively stable when the dimension size is greater than 128. Next, we turn to analyze the effect of the category size of job transition features. As shown in Figure 7, the performance of Ahead is stable on three prediction tasks. Indeed, more categories can better distinguish features, while the embedding of each category is jointly learned, which can also distinguish features automatically. When the category size is greater than 5, Ahead is sufficient to represent the job transition features and get stable performance.

5.2.4 Robustness Analysis. Afterwards, we turn to demonstrate the robustness of Ahead. We explore how the performance of AHGN is

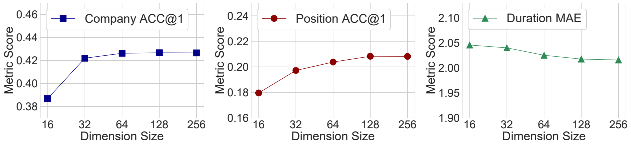


Figure 6: Effect of different input dimension of Dual-GRU.

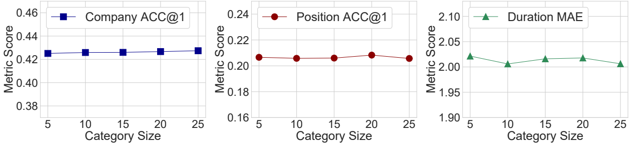


Figure 7: Effect of category size of transition features.

Table 2: The performance on randomly split samples.

Proportion	Company		Position		Duration
	ACC@1	MRR	ACC@1	MRR	MAE
10%	0.3186	0.3938	0.1564	0.2487	2.3194
30%	0.4155	0.4883	0.1947	0.2992	2.2433
50%	0.4224	0.4963	0.2003	0.3072	2.0381
70%	0.4256	0.5004	0.2032	0.3119	2.0354
90%	0.4267	0.5019	0.2073	0.3167	2.0261

Table 3: Top-5 companies and positions that give the highest attention to previous companies and positions respectively.

Companies that pay most attention to former companies.
Ernstandyoung, Pricewaterhouse Coopers Deloitte, IBM, Accenture
Positions that pay most attention to former positions.
Team Lead, Assistant Manager, Project Manager Business Analyst, Account Manager

influenced by the different training ratios. As shown in Table 2, it is obvious that with increasing training proportion, the performance is improved as well. When the training ratios are greater than 0.3, the performance improves slowly. All results are stable, which demonstrates the robustness of our model.

5.2.5 Case Study. With the attention mechanism, we conducted several case studies on job mobility.

At first, we investigated the importance of former companies and positions to the next job mobility based on Equation 14. Figure 8 shows the attention distribution among former companies and positions. When predicting the next company, the previous company would obtain more attention than the previous position. While the situation is opposite when predicting the next position, which may indicate that to join the dream company, employees should choose a more suitable former company. However, to get a dream job position, employees should work on related former positions. Moreover, Table 3 reports the top-5 companies and positions that give the highest attention to the previous company and positions. This may indicate that accounting and consulting companies attach great importance to the former company, and management positions attach great importance to the former positions.

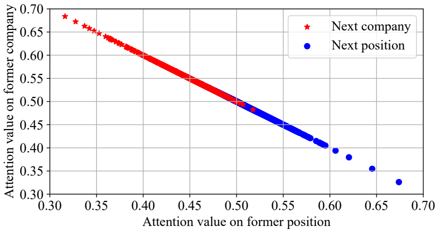
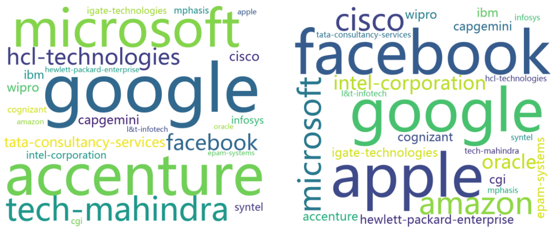


Figure 8: The attention distribution of former company and position in job transition.



(a) Company Frequency (b) Attention Value

Figure 9: The distribution comparison of several companies.

Finally, we analyzed the job transition on a specific job position, i.e.,*software engineer*. We first selected all job transition records in which the former job position is *software engineer*. Then we grouped these records by the former company. Afterwards, we evaluated the attention value for the former company in each record and obtained the mean attention value of each company. Figure 9(a) and Figure 9(b) show the companies that appear most frequently in records and get the highest attention values respectively, where the size of name is proportional to the value. Obviously, the two distributions are quite different, the companies with high frequency may not get high attention. The high-tech companies such as Facebook, Google and Apple get the highest values, which may indicate they are pretty competitive in the position of *software engineer*.

6 CONCLUSION

In this paper, we studied the problem of job mobility prediction by exploring the impact of macro-level job transition relationships. Specifically, we first constructed a heterogeneous company-position network from the massive career trajectory data and then proposed a prediction framework, namely Ahead, based on the attentive heterogeneous graph embedding. In particular, an attentive heterogeneous graph embedding (AHGN) model in Ahead was designed to learn the comprehensive representation of companies and positions. Moreover, the other module in Ahead, namely Dual-GRU model, was applied for individual career path mining with the consideration of the mutual influence between company and position. Finally, extensive experiments conducted on a real-world dataset clearly validated the effectiveness of the proposed framework.

7 ACKNOWLEDGMENTS

This research was partially supported by grants from the National Natural Science Foundation of China (Grant No.61836013, 91746301, 62072423) and the National Key Research and Development Program of China (Grant No.2018YFB1402600).

REFERENCES

[1] William J Anderson. 2012. *Continuous-time Markov chains: An applications-oriented approach.* Springer Science & Business Media.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Mikhail Belkin and Partha Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering.. In *Nips*, Vol. 14. 585–591.

[4] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[5] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 119–128.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[7] 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).

[8] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.

[9] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 273–278.

[10] 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.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[13] Peter A Heslin. 2005. Conceptualizing and evaluating career success. *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 26, 2 (2005), 113–136.

[14] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6, 02 (1998), 107–116.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] 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.

[17] Zhipeng Huang and Nikos Mamoulis. 2017. Heterogeneous information network embedding for meta path based proximity. *arXiv preprint arXiv:1701.05291* (2017).

[18] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

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

[20] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

[21] Huayu Li, Yong Ge, Hengshu Zhu, Hui Xiong, and Hongke Zhao. 2017. Prospecting the career development of talents: A survival analysis perspective. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 917–925.

[22] 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.

[23] Hao Lin, Hengshu Zhu, Yuan Zuo, Chen Zhu, Junjie Wu, and Hui Xiong. 2017. Collaborative company profiling: Insights from an employee's perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[24] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 4823–4832.

[25] Hao Liu, Yongxin Tong, Jindong Han, Panpan Zhang, Xinjiang Lu, and Hui Xiong. 2020. Incorporating Multi-Source Urban Data for Personalized and Context-Aware Multi-Modal Transportation Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[26] Hao Liu, Qiyu Wu, Fuzhen Zhuang, Xinjiang Lu, Dejing Dou, and Hui Xiong. 2021. Community-Aware Multi-Task Transportation Demand Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 320–327.

[27] Junhua Liu, Chu Guo, Yung Chuen Ng, Kristin L Wood, and Kwan Hui Lim. 2019. IPOD: Corpus of 190,000 industrial occupations. *arXiv preprint arXiv:1910.10495* (2019).

[28] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1053–1058.

[29] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206* (2014).

[30] 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.

[31] Thomas WH Ng, Kelly L Sorensen, Lillian T Eby, and Daniel C Feldman. 2007. Determinants of job mobility: A theoretical integration and extension. *Journal of Occupational and Organizational Psychology* 80, 3 (2007), 363–386.

[32] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1105–1114.

[33] Yiming Pan, Xuefeng Peng, Tianran Hu, and Jiebo Luo. 2017. Understanding what affects career progression using Linkedin and Twitter data. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2047–2055.

[34] Karl Pearson. 1895. VII. Note on regression and inheritance in the case of two parents. *proceedings of the royal society of London* 58, 347-352 (1895), 240–242.

[35] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[36] Kristen M Shockley, Heather Ureksoy, Ozgun Burcu Rodopman, Laura F Poteat, and Timothy Ryan Dullaghan. 2016. Development of a new scale to measure subjective career success: A mixed-methods study. *Journal of Organizational Behavior* 37, 1 (2016), 128–153.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[38] Jian Wang, Yi Zhang, Christian Posse, and Anmol Bhasin. 2013. Is it time for a career switch?. In *Proceedings of the 22nd international conference on World Wide Web*. 1377–1388.

[39] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S Yu. 2020. A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources. *arXiv preprint arXiv:2011.14867* (2020).

[40] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.

[41] 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.

[42] Ye Xu, Zang Li, Abhishek Gupta, Ahmet Bugdayci, and Anmol Bhasin. 2014. Modeling professional similarity by mining professional career trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1945–1954.

[43] 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.

[44] Le Zhang, Tong Xu, Hengshu Zhu, Chuan Qin, Qingxin Meng, Hui Xiong, and Enhong Chen. 2020. Large-Scale Talent Flow Embedding for Company Competitive Analysis. In *Proceedings of The Web Conference 2020*. 2354–2364.

[45] 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.

[46] Sheng Zhou, Jiajun Bu, Xin Wang, Jiawei Chen, and Can Wang. 2019. HAHE: Hierarchical attentive heterogeneous information network embedding. *arXiv preprint arXiv:1902.01475* (2019).

[47] Hengshu Zhu, Hui Xiong, Fangshuang Tang, Qi Liu, Yong Ge, Enhong Chen, and Yanjie Fu. 2016. Days on market: Measuring liquidity in real estate markets. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 393–402.

A APPENDIX

A.1 Data Pre-processing.

The original dataset contains more than 400 million professional resumes, which is noisy and inefficient for model training. To filter this dataset, we firstly applied the method mentioned in [45] to extract the career trajectory data. Specifically, if the absolute difference between the end time of the previous job and the start time of the next job is less than a predefined threshold, the job transition is considered valid. Then we constructed a career tree and chose the longest path as individual career trajectory. To unify the messy job title, we firstly extracted the corresponding *responsibility* and *function* words according to the manually annotated IPOD dataset [27]. As a job title describes the responsibilities and function of the job, two job titles can be regarded as the same if they have the same responsibilities and function. Therefore we can aggregate the job titles according to the selected key words. Afterwards, we chose the most frequent companies from different types, and kept the most frequent job titles of them. Finally, we retained the career trajectories among the selected companies and positions after January 2010. Totally, 459,309 career trajectories are extracted, which consist of 1,380 companies and 2,098 job titles.

Further, we also collected the company-specific features, position-specific features and person-specific features, as shown in Table 4. We processed the data with the following methods. For the free text features, such as the company description, we used the *doc2vec* [20] model to transform the text to a fixed-length vector. For the categorical features, such as company type, we employed the one-hot embedding method. For the duration features, we segmented them by every half year, then the one-hot embedding method was applied.

A.2 Evaluation Metrics Description

For next company and position prediction tasks, we selected *Accuracy@k* (*Acc@k*) and *Mean Reciprocal Rank* (*MRR*) to evaluate performance, which are defined as:

$$Acc@k = \frac{1}{N}\sum_{i=1}^{N} I(rank(i) \leq k), \quad MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank(i)}, \tag{18}$$

where N is total number of samples, $rank(i)$ stands for the real rank in the predicting ranking list. $I(\cdot)$ is the indicator function that equals to 1 if $rank(i) \leq k$ and equals to 0 otherwise. Here we set $k = 1, 15, 30$ respectively. The higher values of *Acc@k* and *MRR* means better prediction results. For duration prediction, we adopted *Root Mean Square Error* (*RMSE*) and *Mean Absolute Error* (*MAE*) as evaluation metrics, which are defined as:

$$RMSE(d, \hat{d}) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(d_i - \hat{d}_i)^2}, \quad MAE(d, \hat{d}) = \frac{1}{n}\sum_{t=1}^{n}\left|d_i - \hat{d}_i\right|, \tag{19}$$

where d_i and $\hat{d}_i$ are the predicted duration and the real duration, and the lower values means better performance.

A.3 Baseline Description

We compared our Ahead model with the following methods to predict the job mobility.

- **LR**: It is a supervised model. It fits the samples in the multi-dimensional space by using a linear combination of features.

Table 4: The description of features in our dataset.

Entity	Category	Feature
User	Categorical	User ID
	Text	Self description
Company	Categorical	Company ID
		Company age
		Company type
		Company location
		Company size
	Text	Company description
	Numerical	Company duration
Position	Categorical	Position ID
	Numerical	Position duration

- **RF** [4]: It is an ensemble learning method by constructing a number of decision trees at training time and making prediction according to the individual trees at test time.
- **LSTM** [15]: It is a variant of RNNs, which is proposed to address the vanishing gradient problems by introducing several gates in neural cells. Here we input the company and position feature sequences together and trained three models for three tasks respectively.
- **GRU** [7]: It is also a variant of RNNs for dealing with the vanishing gradient problems. Here the experimental setting was the same as LSTM.
- **NEMO** [22]: It is an encoder-decoder architecture. The encoder maps the multiple heterogeneous profile contexts into a fixed-length vector and the decoder maps the context vector to a sequence of company and position. We modified it by feeding the hidden state in each timestamp into a fully-connected layer to predict the working duration.
- **HCPNN** [30]: It proposes a hierarchical career-path-aware neural network to handle the dynamic nature of career paths for employees. The basic model can only predict the next company as well as the current working duration, we modified it by exchanging the hierarchy of company and position to predict the next position.
- **HAN** [40]: It is a state-of-art heterogeneous graph neural network, which employs node-level attention and semantic-level attention. We modified it by adding the Dual-GRU module, and the output of HAN was the input of Dual-GRU.
- **HGAT** [24]: It proposes a heterogeneous graph attention embedding method for short text classification based on a dual-level attention mechanism. Here the experimental setting was the similar to HAN.

A.4 Parameter setting

In our experiments, the output dimension of external and internal aggregators was set to 128. The dimension size of duration embedding and individual feature embedding was set to 128. And the input and hidden size of Dual-GRU were set to 128 and 256 respectively. In the training phase, we randomly initialized parameters and optimized the model with Adam. We set the mini-batch size to 1024, the hyper-parameters λ_1, λ_2 and λ_3 to 3, 1 and $1e-6$. And the learning rate was set to $1e-3$ with decay as 0.375 every 5 epochs.