NEMO: Next Career Move Prediction with Contextual Embedding

Liangyue Li
Arizona State University
liangyue@asu.edu

How Jing
LinkedIn
hjing@linkedin.com

Hanghang Tong
Arizona State University
hanghang.tong@asu.edu

Jaewon Yang
LinkedIn
jeyang@linkedin.com

Qi He
LinkedIn
qhe@linkedin.com

Bee-Chung Chen
LinkedIn
bchen@linkedin.com

ABSTRACT

With increased globalization and labor mobility, human resource reallocation across firms, industries and regions has become the new norm in labor markets. The emergence of massive digital traces of such mobility offers a unique opportunity to understand labor mobility at an unprecedented scale and granularity. While most studies on labor mobility have largely focused on characterizing macro-level (e.g., region or company) or micro-level (e.g., employee) patterns, the problem of how to accurately predict an employee's next career move (which company with what job title) receives little attention. This paper presents the first study of large-scale experiments for predicting next career moves. We focus on two sources of predictive signals: *profile context matching* and *career path mining* and propose a contextual LSTM model, `NEMO`, to simultaneously capture signals from both sources by jointly learning latent representations for different types of entities (e.g., employees, skills, companies) that appear in different sources. In particular, `NEMO` generates the contextual representation by aggregating all the profile information and explores the dependencies in the career paths through the Long Short-Term Memory (LSTM) networks. Extensive experiments on a large, real-world LinkedIn dataset show that `NEMO` significantly outperforms strong baselines and also reveal interesting insights in micro-level labor mobility.

Keywords

Career move; contextual LSTM; embedding

1. INTRODUCTION

With increased globalization and labor mobility, human resource reallocation across firms, industries and regions has become the new norm in labor markets [14]. Such labor flow is a vehicle that matches supply with demand, stimulates circulation of knowledge at the regional and international scale and proves to be a forceful driver of innovation [4]. Given the large-scale digital traces of labor flows available on the web (e.g., LinkedIn), it is of considerable interest in understanding the dynamics of employees' career moves and the implications on the economy at both the micro and aggregate levels.

Previous studies on labor mobility have mainly focused on either the macro-level analysis (e.g., regions, companies) or characterizing micro-level (e.g., individuals) labor mobility patterns. At the macro level, the employer-to-employer flows are discovered to be procyclical and concentrated among frequent job changers [2]. The *labor flow network* is proposed to identify firms with high growth potential through the lens of network science [14]. At the micro level, the career moves of scientists across institutions are analyzed, and how the moves shape and affect individual's performance is quantified with scholarly data [6]. In the recommendation domain, job recommendation models [1] are built based on whether a user clicks or applies for the recommended jobs. The problem of how to predict an individual's actual next job position (not whether he/she clicks or applies for a job) received relatively little attention [26]. In this paper, we present the first large-scale analysis for predicting an individual's next career move (which company with what job title) for millions of users. By modeling each individual differently, we achieve better prediction accuracy and are able to provide personalized recommendations for each individual from a perspective different from existing job recommendation models [1].

Building personalized predictive model for individual's career is a challenging problem because there can be many factors behind a career move, e.g., education background, skill set, or previous job history and so on. In this paper, we focus on two types of signals that are available on LinkedIn: First (*profile context matching*), the predicted next career move should reflect an individual's profile information, e.g., skills, education, etc, otherwise the so called skills gap would come in the way. An experienced engineer might find it difficult to be competent for an accountant position. The profile attributes can also mitigate the cold-start problem where we do not observe any career history for new users. Second (*career path mining*), the predicted next career move should reflect the trajectory of one's own past career path. The knowledge and experience accumulated along the way prepares a job seeker for the next move and it is very rare for one to switch to an entirely new field.

To build a predictive model using the profile attributes and career paths, the main challenge is how to integrate

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054200

these heterogeneous signals. On the member profile side, we have categorical attributes that are high dimensional, e.g., there are millions of companies but more than half of them have less than 50 employees. Moreover, some attributes are single-valued per member (e.g., final education), while other attributes are multi-valued (e.g., skill set). On the career path side, we have a sequence of job positions (i.e., company and job title). A comprehensive model that can handle both signals is needed.

To simultaneously capture the two types of signals, we propose a contextual LSTM model, named NEMO, inspired by the huge success of neural networks in the several areas (e.g. speech recognition [9] and natural language processing [21]). The proposed model follows the encoder-decoder architecture that can learn effective latent representations/embeddings[1] for the objects (e.g., skills, companies). In particular, the *encoder* maps multiple heterogeneous profile attributes into a fixed-length *context vector*. Concretely, the model first generates the representation for the employee's skill sets by aggregating the embeddings of the skills that the employee has, and then further aggregates the skill set representation with that of the employee's education and location representations. The resulting combined representation would be the employee's context vector. The *decoder*, on the other hand, maps the context vector to the employee's sequence of positions. We take advantage of the Long Short-Term Memory (LSTM) recurrent neural network [10] to pass along the long-term dependencies from the previous positions. Specifically, the employee's context vector is used as the initial state of the LSTM network to generate the career path. The hidden states in LSTM capture not only the contextual information, but also the dynamics along one's career path. LSTM is a natural fit in our setting, due to its proven capability in forming implicit compositional representations over sequences [7].

We conduct the first large-scale experiments for predicting career moves of individuals using a dataset with millions of LinkedIn members. Our experiments show two findings. First, by using signals from both the profile context and career path, we achieve significantly better performance than a number of strong baselines for predicting the next career move. In particular, we empirically show that each of these signals is crucial for making accurate predictions. Second, the model which is trained end-to-end without injecting any prior knowledge uncovers insightful patterns from our large-scale analysis.

The main contributions of this paper are as follows:

1. **Problem Formulation:** We formally define the Next Career Move Prediction, to predict an employee's next career move, i.e., his/her next company and title. To the best of our knowledge, this is the first study of large-scale analysis for predicting next career move.

2. **Algorithm and Analysis:** We propose NEMO, a contextual LSTM model that integrates the *profile context* as well as *career path* dynamics.

3. **Empirical Evaluations:** We conduct extensive experiments on a real-world, large-scale LinkedIn dataset with millions of users and demonstrate the superiority of our model compared to several strong state-of-the-art baselines.

4. **Qualitative Insights:** We draw interesting insights from the prediction case studies as well as career path sampling.

Table 1: Table of symbols

Symbols	Definition
$\mathcal{J}^u = \{J^u_1, J^u_2, \ldots, J^u_n\}$	user u's working experience
$J^u_i = (l^u_i, c^u_i, t^u_i)$	user u worked at company c^u_i with title l^u_i starting from time t^u_i
$\mathcal{S}^u = \{s_1, s_2, \ldots, s_m\}$	user u's skills set
h^u	user u's education institute
r^u	user u's current location
$\mathcal{U}$, $\mathcal{L}$, $\mathcal{C}$, $\mathcal{K}$, $\mathcal{H}$, $\mathcal{R}$	the collections of all users, titles, companies, skills, schools, locations

The rest of the paper is organized as follows. Section 2 formally defines Next Career Move Prediction. Section 3 proposes our model. Section 4 presents the experimental results. Section 5 reviews the related work and the paper concludes in Section 6.

2. PROBLEM DEFINITION

In this section, we present the notations used throughout the paper (summarized in Table 1), and formally define the Next Career Move Prediction problem.

LinkedIn is the world's largest professional network where members can create their professional profiles and seek jobs. Users can share their working experience by reporting the employers they have worked for. Specifically, a user u's working experience can be summarized as $\mathcal{J}^u = \{J^u_1, J^u_2, \ldots, J^u_n\}$, where J^u_i is user u's i-th job position, denoted by a tuple, i.e., $J^u_i = (l^u_i, c^u_i, t^u_i)$, indicating that user u worked at company c^u_i with title l^u_i starting from time t^u_i. Besides the working experience, users can also add skills on the profile or get their skills endorsed. For example, a user might be good at *Data Mining, Machine Learning* and *Pattern Recognition*. We denote user u's skills set by $\mathcal{S}^u = \{s^u_1, s^u_2, \ldots, s^u_m\}$, where each s^u_i is a specific skill, e.g., *Hadoop*. The user can also report their education background in their profile. The user's location (e.g., San Francisco Bay Area) is denoted by r^u. For simplicity, we only consider user u's highest education institute and denote it by h^u. Let $\mathcal{U}$, $\mathcal{L}$, $\mathcal{C}$, $\mathcal{K}$, $\mathcal{H}$ and $\mathcal{R}$ be the collections of all users, titles, companies, skills, schools and locations, i.e., $u \in \mathcal{U}$, $l^u_i \in \mathcal{L}$, $c^u_i \in \mathcal{C}$, $\mathcal{S}^u \subseteq \mathcal{K}$, $h^u \in \mathcal{H}$ and $r^u \in \mathcal{R}$. Note that all the entities (e.g., titles, companies) are standardized, e.g., the two different titles *Senior Software Engineer* and *Sr. Software Engineer* are mapped to the same item in $\mathcal{L}$. In the paper, we use bold lowercase letters for vectors, e.g., we use $\mathbf{s}^u_1, \ldots, \mathbf{s}^u_m$ to denote the embedding vectors of skills $s^u_1, \ldots, s^u_m$, and bold upper-case letters for matrices (e.g., $\mathbf{W}$). Also, we represent the elements in a matrix using a convention similar to Matlab, e.g., $\mathbf{W}(:, j)$ is the j^{th} column of $\mathbf{W}$, etc.

With the above notations, the problem of predicting each individual's next career move can be formally defined as follows:

Problem 1. Next Career Move Prediction

Given: *the working experience of all users* $\mathcal{J}^{u_1}, \ldots, \mathcal{J}^{u_{|\mathcal{U}|}}$ *observed up to a timestamp* T*, the skills sets* $\mathcal{S}^{u_1}, \ldots, \mathcal{S}^{u_{|\mathcal{U}|}}$,

[1] We use representation and embedding interchangeably in the paper.

the education institutes $h^{u_1}, \dots, h^{u_{|\mathcal{U}|}}$ *and the locations* $r^{u_1}, \dots, r^{u_{|\mathcal{U}|}}$ *of all users*

Predict: *the user* u*'s next career move, including title* $l_{|\mathcal{J}^u|+1} \in \mathcal{L}$ *and company* $c_{|\mathcal{J}^u|+1} \in \mathcal{C}$*, after time* T*.*

As an illustrative example, Figure 1 shows one LinkedIn member's working experience. We can see that the member worked as a research staff member at IBM Almaden Research Center from Dec. 2010 to June 2013. Suppose we can observe the member's career history up to June 2013, the problem is to predict the member's next title and company after June 2013, that is, staff researcher at LinkedIn in this case.

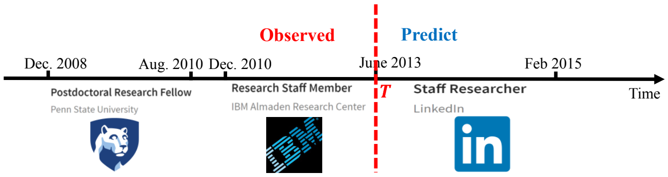

Figure 1: One LinkedIn member's working experience.

3. PROPOSED SOLUTIONS

In this section, we present our solution for Problem 1. We start with the design objectives for the Next Career Move Prediction, and then present the details of our `NEMO` predictive model, followed by the model learning.

3.1 Design Objectives

From the prediction perspective, we focus on leveraging all the information available in the user's LinkedIn profile. To be specific, we want to achieve the following two design objectives:

- *Profile context matching.* The three most salient sections in LinkedIn members' profile that are indicative of users' career are *Skills*, *Education* and *Location*. Skills are a critical asset for individuals and different jobs have different requirements for skills. Matching skills and jobs has become a high-priority policy concern. For instance, an individual with strong skills in machine learning and data mining is more likely to move to a research scientist position in a high tech company than an accountant position in a bank. The next important attribute we consider is education. A Carnegie Mellon University graduate might have a higher chance to work in the tech industry compared to a university best known for its law major. Last but not least, location of job seekers also biases where they would eventually go. Companies in the Bay Area are generally more attractive to Bay Area job seekers compared to New York based companies. Being able to incorporate all these contextual information proves to be the key to match top talents and companies. For simplicity, we assume that all these profile attributes are static and fixed.
- *Career path mining.* Another important signal for the career move is user's current position. It is natural to assume that the next position is highly correlated to the current one and existing work incorporating this information show great improvement upon static methods [28, 33]. However, most professionals might have more than one jobs throughout their career history, and considering only the last position misses the bigger picture of one's professional life. It is the knowledge and the experiences built up through one's entire career history that prepares the candidate for future opportunities. It is thus desirable for us to learn the entire career trajectory in order to infer the next move.

In Section 4, we will show that each of these design objectives is so crucial in predictive performance that we need to incorporate both in the model.

3.2 NEMO - An Encoder-Decoder Architecture

We carefully design our predictive model to fulfill the above objectives. The proposed model follows the encoder-decoder architecture depicted in Figure 2. The encoder maps the multiple heterogeneous profile contexts into a fixed-length vector, which we refer to as *context vector* and the decoder maps the context vector to a sequence of positions. Such neural network model is able to compute the conditional probability of the output career path given the input user's profile information. Success of similar framework has been shown in machine translation where one encodes a source language sequence to a vector and then decodes to the target language sequence [29].

3.2.1 Encoding the Context Representation

We want to learn a compact representation of the context information from users' profile. The attributes are usually high-dimensional categorical features that bear no notion of similarity and do not generalize well. One way to encode these discrete entities is to use embeddings inspired by the success of distributed representations of words in natural language processing tasks [23].

We now explain how we construct embedding for user's profile attributes and begin with skill embedding. Let $\{\mathbf{s}_1^u, \mathbf{s}_2^u, \dots, \mathbf{s}_m^u\}$ be the set of user u's skills embeddings. Note that each user has different number of skills, from as few as one skill to more than 10 skills. To ensure that each user has embedding with the same dimension, we use a pooling method across skill embedding vectors. In particular, we perform max-pooling to get a single skill vector,

$$\mathbf{s}^u = \max(\mathbf{s}_1^u, \mathbf{s}_2^u, \dots, \mathbf{s}_m^u), \tag{1}$$

where $\max(\cdot)$ is applied dimension-wise. The intuition is that a user's top skills might dominate more in the career move. Note that we also tested other pooling methods such as average pooling but found that max-pooling performs the best, which aligns with the finding in [33]. Next attributes are user's school and location. We concatenate their embeddings with skill embedding, and feed it through a one-layer neural network as follows:

$$\mathbf{v}_u = \tanh\Big(\mathbf{W}_v[\mathbf{s}^u, \mathbf{h}^u, \mathbf{r}^u]^T + \mathbf{b}_v\Big), \tag{2}$$

where $\mathbf{h}^u$ and $\mathbf{r}^u$ are the user's school and location embeddings, $[\cdot, \cdot, \cdot]$ concatenates the vectors and $\mathbf{W}_v$ and $\mathbf{b}_v$ are the projection matrix and bias vector, respectively. The final output $\mathbf{v}_u$ from the encoder captures the correlations of

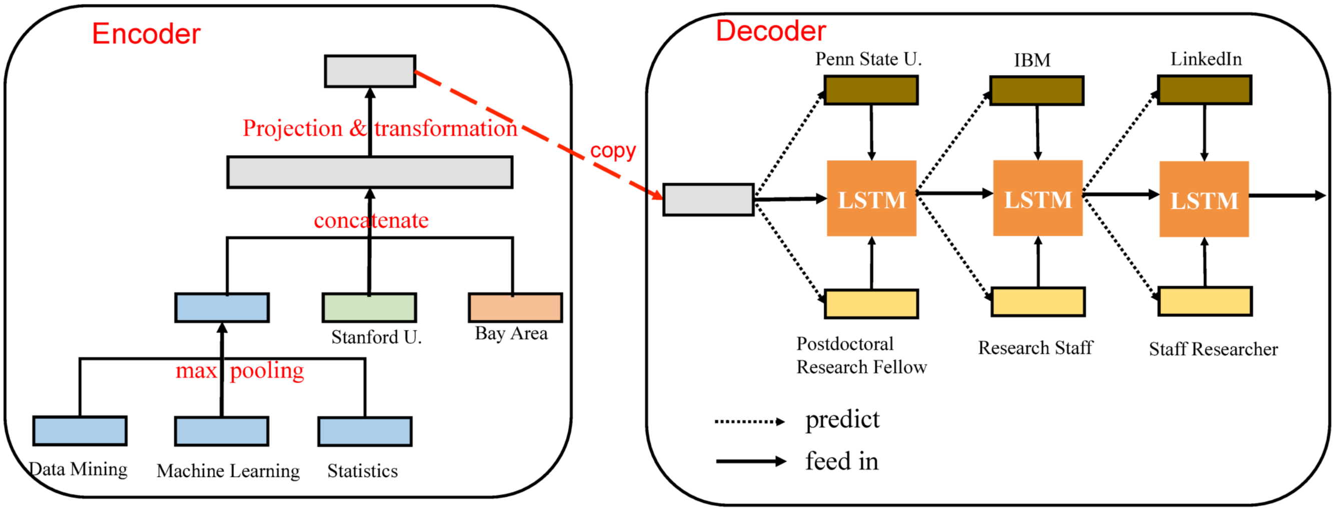

Figure 2: Framework of NEMO.

user's skills, school, location and serves as the representation of user's profile. We call vector $\mathbf{v}_u$ the *context vector* for user u.

3.2.2 *Decoding with Long-Short Term Memory Networks*

In order to decode one's career path from the context vector learned, we take advantage of the Recurrent Neural Network (RNN) due to its success in modeling sequential data [24, 3]. In particular, we employ a Long short-term memory (LSTM) [15], as a particular type of RNN, which was proposed to address the problem of vanishing gradients. LSTM is capable of exploiting longer range of temporal dependencies in the sequences and has been the state-of-the-art for several tasks, including sequence to sequence learning [29], image caption generation [31]. There are many variants of LSTM architecture and we refer interested readers to [12]. In this paper, we use the following LSTM equation:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{im}\mathbf{m}_{t-1})\\
\mathbf{f}_t &= \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fm}\mathbf{m}_{t-1})\\
\mathbf{o}_t &= \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{om}\mathbf{m}_{t-1})\\
\mathbf{e}_t &= \mathbf{f}_t \odot \mathbf{e}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{cm}\mathbf{m}_{t-1})\\
\mathbf{m}_t &= \mathbf{o}_t \odot \mathbf{e}_t
\end{aligned} \tag{3}$$

where $\odot$ is the element-wise multiplication, $\mathbf{x}_t$ is the input data, i.e., the embeddings of company and title at time step t; $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$ serve as *input gate, forget gate, output gate* respectively, the various $\mathbf{W}$ matrices are the trained parameters, and $\mathbf{m}_t$ is the hidden state at time step t. The hidden state vector $\mathbf{m}_t$ can be viewed as the dynamic representation of a user at time t that aggregates the user's profile context as well as the user's career history up to t.

3.3 NEMO - Learning and Prediction

Our final architecture is a combined encoder-decoder network and the entire model is trained end-to-end to maximize the log probability of generating the correct career path given the observed users' context information:

$$\theta^* = \arg\max_{\theta} \sum_{u \in \mathcal{U}} \log p(\mathcal{J}^u | \mathcal{S}^u, h^u, r^u), \tag{4}$$

where θ are all the model parameters, including all the entities' embeddings and parameters in LSTM. Suppose we have observed a particular user u's T jobs, the log probability for that user's career path further decomposes into,

$$\begin{aligned}
&\log p(\mathcal{J}^u | \mathcal{S}^u, h^u, r^u)\\
&= \sum_{t=1}^{T} \log p(J_t^u | \mathcal{S}^u, h^u, r^u)\\
&= \sum_{t=1}^{T} \big[\log p_t(c_t^u | \mathcal{S}^u, h^u, r^u, c^u_{t'<t}, l^u_{t'<t})\\
&+ \log p_t(l_t^u | \mathcal{S}^u, h^u, r^u, c^u_{t'<t}, l^u_{t'<t})\big]
\end{aligned} \tag{5}$$

where $c^u_{t'<t}$ and $l^u_{t'<t}$ are the user's previous $t-1$ companies $c_1^u, \ldots, c_{t-1}^u$ and $t-1$ titles $l_1^u, \ldots, l_{t-1}^u$. Note that we assume c_t^u and l_t^u are conditionally independent of each other given everything observed up to time $t-1$. To get the probability distribution $p_t(c_t^u | \mathcal{S}^u, h^u, r^u, c^u_{t'<t}, l^u_{t'<t})$ over companies, we use the hidden states vector $\mathbf{m}_{t-1}$ from the LSTM in Eq. (3) and feed it to a *softmax* layer, i.e.

$$\begin{aligned}
&p_t(c_t^u = k | \mathcal{S}^u, h^u, r^u, c^u_{t'<t}, l^u_{t'<t}) =\\
&\frac{\exp(\mathbf{W}_c(:,k)^T \mathbf{m}_{t-1} + \mathbf{b}_c(k))}{\sum_{c' \in \mathcal{C}} \exp(\mathbf{W}_c(:,c')^T \mathbf{m}_{t-1} + \mathbf{b}_c(c'))}
\end{aligned} \tag{6}$$

where $\mathbf{W}_c, \mathbf{b}_c$ are the softmax weight, bias for company respectively. Similarly we feed $\mathbf{m}_{t-1}$ to another *softmax* to predict title distribution. In other words, we are doing multi-task learning to predict the next company and title jointly with a shared representation $\mathbf{m}_{t-1}$.

However, it would be practically infeasible to directly maximize the log probability in Eq. (4) since computing the full *softmax* would have a cost proportional to the number of companies and titles, which are usually very large, e.g., there are in the order of millions of companies in U.S. alone and hundreds of thousands even after aggressive preprocessing. To improve scalability, we adopt the "sampled *softmax*" strategy to approximately maximize Eq. (4). The basic idea is instead of performing *softmax* over the entire output space, we randomly sample a subset (e.g. 50) of companies/titles and do the *softmax* over this much smaller space. We omit details here for space limit and refer interested readers to [17] for more rigorous derivations.

After learning, it becomes straightforward to predict the user's next career move. Suppose we have observed a user u's career path until time T, and want to predict what u's next company and title would be. We can first obtain the hidden states vector $\mathbf{m}_T$, which captures all the contextual information and the career path dynamics up until time T. We then predict the next company and title using the **full** *softmax* to get the full distribution over the next company, title and select the top-K most probable results.

3.4 Discussion

We will show in Section 4 that NEMO gives superior predictive performance. In addition to the predictive power, however, we also note that our model allows us to *sample* career trajectories from a given member profile. In other words, our model essentially defines probability distribution of career given the contextual profile. With this generative ability, we can answer questions like "what kinds of career path does a Stanford Computer Science graduate have?". Such insight will be useful for students who are applying for graduate schools. We will show some of sampled career paths in Section 4.4.2.

4. EMPIRICAL EVALUATIONS

In this section, we present the experimental evaluations. The experiments are designed to inspect the following aspects: (1) *Effectiveness:* how accurate are the proposed NEMO model for predicting next career move? and (2) *Insights:* what insights can we draw from the model?

4.1 Dataset

We use the real-world data from LinkedIn to evaluate the proposed model. In particular, we construct two datasets as follows. (1) *Computer*, which consists of members from the following industries: "computer software", "internet", "computer hardware", "computer networking" and "information technology and services"; and (2) *Finance*, which consists of members from the following industries: "banking", "financial services", "investment banking", "investment management". Industries are pre-defined by LinkedIn for users to choose. Both datasets span from the inception of LinkedIn service to 09/24/2016. For preprocessing, we remove members with no positions or with more than 20 positions reported in their profile. We also remove skills, companies, titles and schools that appear less than 10 times in the dataset. The positions, i.e. tuples of company and title, observed up to 12/01/2015 are used for training the model and the task is to predict the first new position (i.e., both company and title) after 12/01/2015. The statistics of the two datasets after preprocessing are summarized in Table 2.

4.2 Experimental setup

Evaluation Metric: We use the Mean Percentile Ranking (**MPR**) [16] to evaluate the quality of the prediction. Let $\mathcal{U}_{test}$ be the set of members who have a new position during the testing period. The MPR for both the company and title prediction can be computed as follows:

Table 2: Statistics for our two datasets. Note that only the scale is reported for the privacy concern.

#members	>1M
#skills	>10K
#companies	>100K
#titles	>10K
#schools	>1K
#locations	>100
#training positions	>10M
#testing positions	>100K

$$MPR(c) = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{|\mathcal{C}|} rank(c_u^*)$$

$$MPR(l) = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{|\mathcal{L}|} rank(l_u^*),$$

where $rank(c_u^*)$ and $rank(c_u^*)$ are the rank of user u's actual company c_u^* and actual title l_u^*, and the rank is obtained by sorting the model's prediction scores. Lower values are more desirable as they indicate the model can rank the true company/title higher in the ranking list. Note that classic classification metrics (precision and recall) are ranking-agnostic, and the Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) at certain ranking position are too coarse given there is only one ground-truth in the ranking list.

Comparison Methods: We compare our NEMO with the following strawmen and state-of-the-arts:

- **Top**: always recommend the most popular company/title.
- **Bigram**: estimate the transition probability using a simple counting method. This is a consistent estimator under the first-order Markov assumption. It is usually a strong baseline when the data is not sparse.
- **Context Only**: use only the contextual information of users without considering the career path to recommend the next position.
- **MC**: Markov Chain sequential model [28] that embeds each company and title into the semantic space and consider only the previous company/title in the prediction phase.
- **HRM**: Hierarchical Representation Model [33] that simply aggregates the embeddings of all the previous companies/titles through max-pooling to make the prediction.
- **LSTM**: use only LSTM to explore the whole career path without the profile context. This model was recently applied in the next item recommendation by [36].
- **NEMO**: the context-aware LSTM model proposed by this paper, which encodes different contextual information from a member into a latent vector representation, and then learns to decode the members' career trajectory based on this vector.

Table 3: Mean percentile rank comparisons on predicting the next title and the next company.

	Computer		*Finance*	
	Company	Title	Company	Title
TOP	0.1318	0.0634	0.1098	0.0663
Bigram	0.1054	0.0437	0.0850	0.0518
Context Only	0.0512	0.0286	0.0403	0.0391
MC	0.0542	0.0277	0.0496	0.0351
HRM	0.0519	0.0269	0.0499	0.0369
LSTM	0.0432	0.0225	0.0411	0.0299
`NEMO`	**0.0299**	**0.0182**	**0.0260**	**0.0253**

Implementation Details: For **Top** and **Bigram**, we randomly recommend a position if multiple positions meet the recommendation criteria. For all neural network-based methods, we use mini-batch SGD with Adagrad acceleration [8], where the batch size is set to 64. The learning rate is set to be 0.05 divided by the batch size. We use small l_2 regularization in each model. The embedding dimension and the number of hidden units are both set to 200, with 2 hidden layer for all the models.

4.3 Quantitative Results

Summarized Results: The performance of each model is presented in Table 3. A salient observation is that our `NEMO` model significantly outperforms all the comparison methods on both datasets. In particular, compared to the best baseline LSTM, we achieve about **30%** and **19%** relative improvements in company and title prediction respectively on *Computer*.

Table 3 also shows the effectiveness of the two important ingredients of our proposed model: profile context and career path. Models incorporating the career path (HRM and LSTM) outperform the models using the last position only (MC and Bigram). Compared with HRM, LSTM performs better because it models the ordering of the positions, whereas HRM simply aggregates the history. Finally, `NEMO` outperforms LSTM, showing the importance of modeling context in addition to the position sequences.

Results with Varying Embedding Dimension: We now compare in more details how varying the embedding dimension affects the performance of each model on *Computer*. From Figure 3, we observe a diminishing return in the performance of all the models. For instance, our `NEMO` with 50 dimensional embeddings performs almost as well as that with 200 dimensions, but enjoys 3 times faster training as well as smaller memory footprint.

Results Segmented by Position's Popularity: Figure 4 presents how performance varies with the popularity of the users' actual company and title in *Computer*. As can be seen from the figure, the improvement of `NEMO` is especially dramatic when the target company/title is really rare, in which case Bigram would fail due to insufficient data for estimating the transition probabilities. On the other end, all models have a small MPR for predicting very popular targets.

Results Segmented by Member's Seniority: The performance of each model with varying members' seniority in *Computer* is shown in Figure 5 where the seniority is defined by the number of positions the member has in the training set.

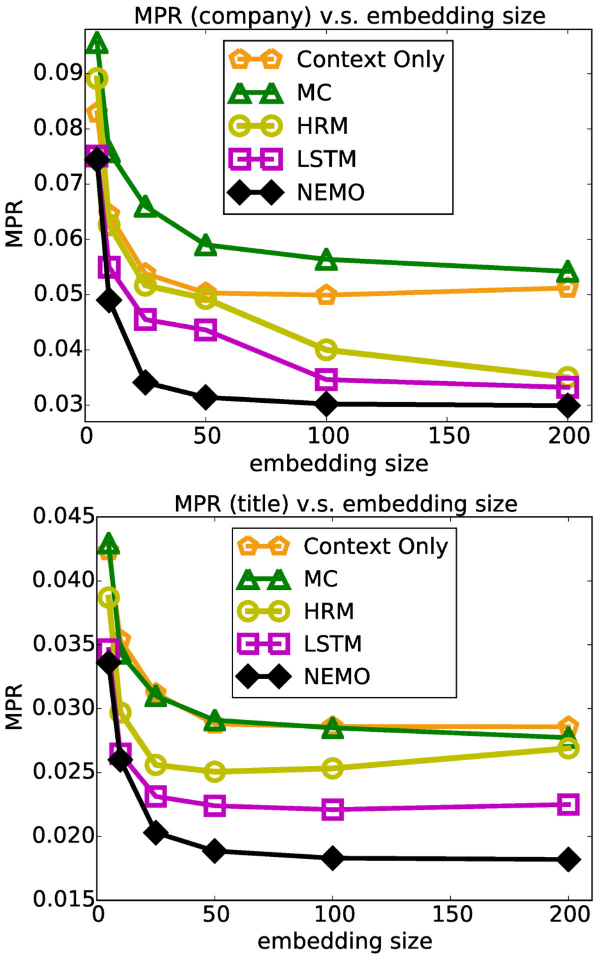

Figure 3: Performance with varying embedding size of the comparison methods on *Computer*. The lower the better. Top: performance on company prediction, Bottom: performance on title prediction.

This experiment shows the importance of out two design objectives: Career path modeling and Profile context matching. First, we focus on the benefit of considering career path. Context only model, which does not use career path at all, shows flat performance regardless of the number of positions observed, while all other methods achieve smaller error as we observe more positions. Moreover, the experiment shows that considering all career positions is better than using the last position only. The models using all positions (HRM, LSTM and `NEMO`) outperform the models using the last position only (MC and Bigram) as a member has more and more positions.

On the other hand, profile context is powerful for users with very few observed positions. We note that baselines using all career positions (LSTM and HRM) do not perform well for members with very few observed positions. For example, when a members has only one position observed, Context-only model outperforms all other baselines in both title prediction and company prediction. Since `NEMO` leverages profile attributes, it can outperforms models solely based on career path significantly when a member has a very short history (i.e., cold-start case).

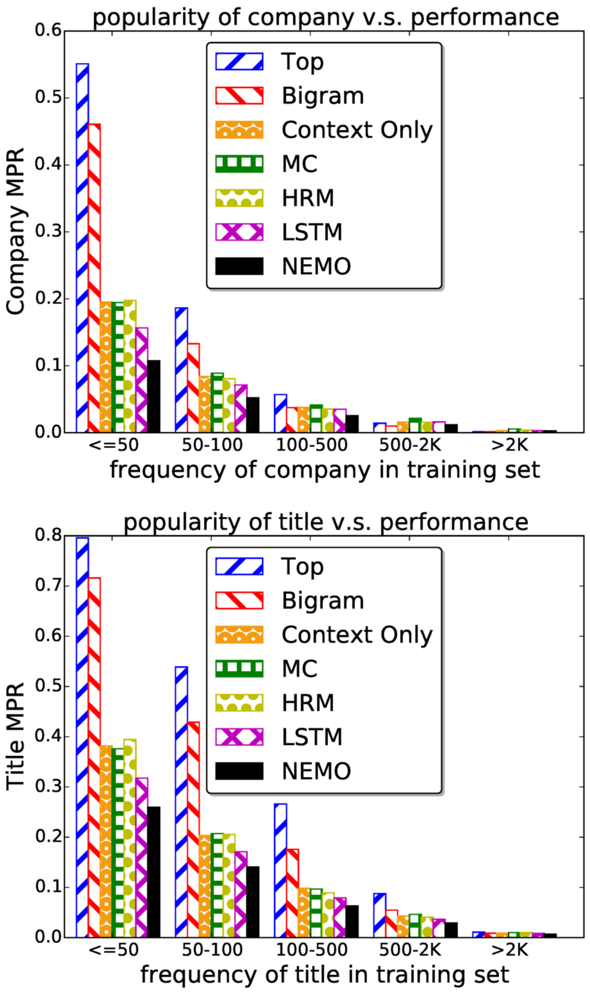

Figure 4: Performance with varying popularity of target company/title in *Computer*. The lower the better. Top: evaluation on company prediction. Bottom: evaluation on title prediction.

4.4 Qualitative Analysis

4.4.1 Prediction Case Study

We present a few anecdotal evidences that show how NEMO predicts next position accurately when other models cannot. Table 4 shows predictions for two members from NEMO and two baseline (Context-only, Bigram). We show the member's previous position (which is given to the model), current position (ground-truth), and the top 5 companies and the top 5 titles predicted by NEMO.

For the user at the top row, he transitioned from a investment company to a airline company, which is very hard to predict. Indeed, Bigram and Context Only models could not get the correct company even at the top 100. We found that the reason for our model to be able to predict correctly is that this user has worked at a Airline company before, and LSTM model was able to "remember" that in the memory cell to make correct recommendations in the future, which was not possible for models that do not consider sequence. Moreover, NEMO leverages that the member has been working in Dallas, which would help narrow down to predicting Southwest Airlines.

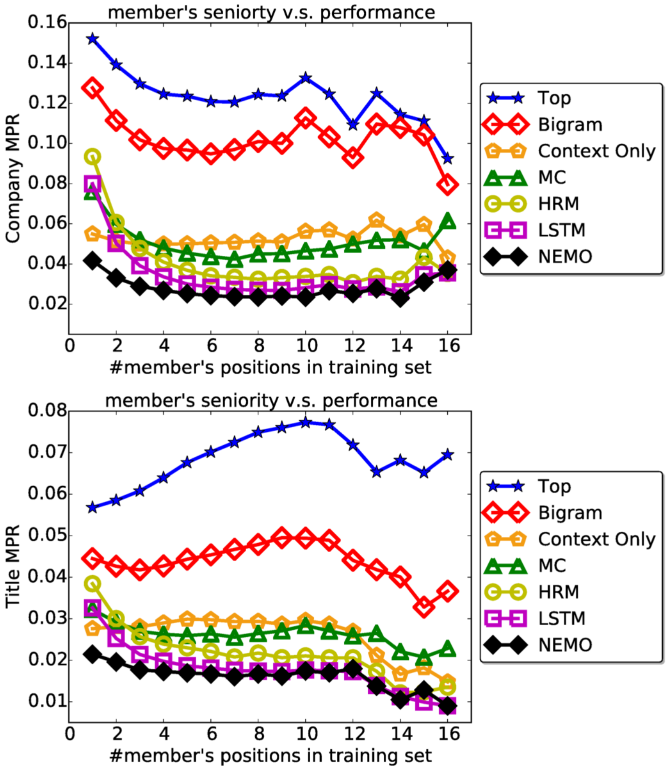

Figure 5: Performance with varying seniority of a member in *Computer*. The lower the better. We stratify members into different buckets by the number of position they have in the past. The average MPR is shown for each bucket. Top: evaluation on company prediction. Bottom: evaluation on title prediction.

For the second user, again, we found that the user has worked at the same company United States Patent and Trademark Office (USPTO) three positions before. Also the user has been working in the Washington, D.C. metro area and has information technology skills. In both cases, NEMO is able to provide accurate predictions due to its power to combine profile context as well as career trajectory.

4.4.2 Sampling Career Path with NEMO

When LSTM is trained on natural text, sampling one word at a time allows us to probe into what the model has learned about the text. In our scenario, the model is trained on professionals' career path and we can sample one position at a time to form a *career trajectory* of a member. Now, suppose the model is input with a member at SF Bay area with skills "machine learning", "data mining", "artificial intelligence" and "algorithms", graduated from Carnegie Mellon University and with first job as *Machine Learning Engineer* at *Google*, we obtain the following sampled path: *Machine Learning Engineer* at *WhatsApp Inc.* → *Machine Learning Engineer* at *Uber* → *Data Scientist* at *Facebook* → *Engineering Lead* at *LinkedIn.* Our model can also handle cold-start users for whom we do not observe any prior working experience. For instance, given a member having skills "Financial Services", "Investments" and graduated from Harvard Business School living in the Greater New York Area, we obtain the following sampled trajectory: *Investment Banker* at *Citi*

Table 4: Prediction case study for 2 users. We list users' previous position, ground-truth next position, top 5 predicted companies and titles from left to right in the table. Bigram model ranks the *Southwest Airlines* at 125th place for the first user (top row) where Context Only method ranks it at 146th. For second user (bottom row), Bigram ranks *USPTO* at 1319 whereas Context Only ranks at it 395.

previous position	Ground-truth next position	Top 5 recommended company	Top 5 recommended Title
Senior Project Manager at *Fidelity Investments*	*Project Manager* at *Southwest Airlines*	Fidelity Investments American Airlines **Southwest Airlines** Epsilon Bank of America	Senior Project Manager **Project Manager** Technical Project Manager Senior Technical Project Manager Program Manager
Software Architect/Tech Lead at *Bureau of Labor Statistics*	*Consultant* at *United States Patent* *and Trademark Office (USPTO)*	Fannie Mae **USPTO** FINRA Lockheed Martin Freddie Mac	Technical Lead Senior Software Engineer **Consultant** Senior Consultant Solutions Architect

→ *Technology Strategist* at *Citi* → *Relationship Manager* at *Citi* → *Vice President* at *Morgan Stanley* → *Vice President Brokerage* at *JPMorgan Chase*. As can be seen, both members have a rising career trajectory. These sampled career trajectories can provide guidance to students in terms of university and major selections. Note that NEMO is the first model that can draw sample career trajectories given members' attributes since it handles both profile context and career sequence.

5. RELATED WORK

In this section, we review related work in terms of (a) labor mobility, and (b) representation learning.

Labor Mobility. Quantifying and modeling labor mobility has been extensively studied in the economics literature. Early work combine a search model with a matching model to identify reasons behind workers' move from job to job as well as move into and out of the labor markets and develop the view that the move is because of changes in the perceived value of workers' market opportunities [18]. The Labor Force Survey data has been examined to establish several key facts regarding the properties of the labor market flows, including the transition probabilities between employment, unemployment and inactivity [11]. Tools from the network science have been brought into economics to characterize the properties of the labor flow network among the different companies and prove to be useful in identifying firms with high growth potential [14]. Thanks to the availability of massive datasets providing individuals' career path, large-scale studies of the labor flow become possible. Academia, as a particular job market, exhibits a unique career movement pattern that is characterized by a high degree of stratification in institutional ranking [6]. The impact of such movement on scientists' research performance has also been quantified. Job recommendation with emphasis on the tenure is effective in improving the utility of the system, i.e., making the recommendation at the right time when the user is likely to change the job is critical [32]. The career trajectories can be employed as professional similarities between two individuals by first aligning the sequences and then extracting the temporal and structural features [34].

[26] is one of the seminal papers on predicting individual's career transition. We note that our work differentiates itself from [26] in that our work leverages a full career trajectory while their is solely based on profile information, and our work conducted a very large scale predictive task with millions of users while they did with less than 100,000 users.

Representation Learning. Representation learning aims to learn good feature representation for input entities without hand-crafting rules. It has shown promising results in many application domains, ranging from natural language processing [23], network science [27] to health care [5]. In NLP, skip-gram model [23] learns embedding for words by predicting a word's surrounding words and the embeddings learned exhibit linguistic regularities that have analogy to algebraic operations [25]. The task of fine grained entity type classification can also be addressed by embedding methods on labels [35]. In computer vision, Multi-model concept representations from the concatenation of linguistic representation vectors and visual concept representation vectors have a substantial performance gain on some semantic relatedness evaluation tasks [19]. The image and sentence embeddings can also be jointly learned in the same space and is shown to be effective for ranking images and descriptions and is able to capture multi-model regularities [20]. Some recent efforts in network science have been devoted to learn embeddings for vertices in a network that can encode the structural relations. DeepWalk [27], in particular, applies skip-gram model to the truncated random walks and achieve improvement on multi-label classification tasks on several social networks. Richer representations can be learned through a biased random walk procedure [13]. LINE [30] learns network embeddings by optimizing a carefully designed objective function that preserves both the first-order and second-order proximities. Several other use cases include representing physical locations with spatial and temporal contexts modeled using a recurrent model for the next location predictions [22] and embedding the dynamics of baskets of items to enhance the performance of next basket recommendation [36].

6. CONCLUSION

In this paper, we study the problem of NEXT CAREER MOVE PREDICTION to predict an employee's next career move. We propose a contextual LSTM model named NEMO that integrates the *profile context* as well as *career path* dynamics. The proposed model follows the encoder-decoder architecture and we show significant improvements over strong baselines. There are many interesting future directions. First, it is desirable to provide interpretable predictions. We are working on attention network to let the model focus on different skills for different positions. Second, user homophily can be exploited from users' social connections. It would be interesting to see how one's career is affected by their close friends or colleagues. Third, our current model assumes that attributes (e.g. skills) are static for simplicity, which might not be true in practice. It would be interesting to model the dynamics of the attributes, resulting in a sequence-to-sequence style model.

7. ACKNOWLEDGMENTS

The 3rd author is partially supported by DTRA HDTRA1-16-0017, NSF IIS-1651203, and ARO W911NF-16-1-0168.

8. REFERENCES

[1] S. T. Al-Otaibi and M. Ykhlef. A survey of job recommender systems. *International Journal of Physical Sciences*, 7(29):5127–5142, 2012.

[2] M. Bjelland, B. Fallick, J. Haltiwanger, and E. McEntarfer. Employer-to-employer flows in the united states: Estimates using linked employer-employee data. *Journal of Business & Economic Statistics*, 29(4):493–505, 2011.

[3] A. Borisov, I. Markov, M. de Rijke, and P. Serdyukov. A neural click model for web search. In *WWW*, 2016.

[4] R. Boschma, R. H. Eriksson, and U. Lindgren. Labour market externalities and regional growth in sweden: The importance of labour mobility between skill-related industries. *Regional Studies*, 48(10):1669–1690, 2014.

[5] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun. Multi-layer representation learning for medical concepts. In *KDD*, 2016.

[6] P. Deville, D. Wang, R. Sinatra, C. Song, V. D. Blondel, and A.-L. Barabási. Career on the move: Geography, stratification, and scientific impact. *Nature Scientific Reports*, 4, 2014.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[8] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(Jul), 2011.

[9] L. D. A. A. George Dahl, Dong Yu. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. volume 20, pages 30–42, January 2012.

[10] F. A. Gers and E. Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001.

[11] P. Gomes. Labour market flows: Facts from the United Kingdom. *Labour Economics*, 19, 2012.

[12] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. *CoRR*, 2015.

[13] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, 2016.

[14] O. A. Guerrero and R. L. Axtell. Employment growth through labor flow networks. *PloS one*, 8(5), 2013.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[16] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[17] S. Jean, K. Cho, R. Memisevic, and Y. Bengio. On using very large target vocabulary for neural machine translation. In *ACL-IJCNLP*, 2015.

[18] B. Jovanovic. Matching, turnover, and unemployment. *The Journal of Political Economy*, 1984.

[19] D. Kiela and L. Bottou. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *EMNLP*, 2014.

[20] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics*, 2015.

[21] J. Li, R. Li, and E. H. Hovy. Recursive deep models for discourse parsing. In *EMNLP*, pages 2061–2069, 2014.

[22] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, 2016.

[23] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013.

[24] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[25] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *NAACL-HLT*, 2013.

[26] I. Paparrizos, B. B. Cambazoglu, and A. Gionis. Machine learned job recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 325–328, New York, NY, USA, 2011. ACM.

[27] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710. ACM, 2014.

[28] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.

[29] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW*, 2015.

[31] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[32] J. Wang, Y. Zhang, C. Posse, and A. Bhasin. Is it time for a career switch? In *WWW*. ACM, 2013.

[33] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, 2015.

[34] Y. Xu, Z. Li, A. Gupta, A. Bugdayci, and A. Bhasin. Modeling professional similarity by mining professional career trajectories. In *KDD*, 2014.

[35] D. Yogatama, D. Gillick, and N. Lazic. Embedding methods for fine grained entity type classification. In *ACL*, 2015.

[36] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In *SIGIR*, 2016.