

A combined representation learning approach for better job and skill recommendation

Vachik S. Dave

Indiana University Purdue University Indianapolis
vsdave@iupui.edu

Mohammad Al Hasan

Indiana University Purdue University Indianapolis
alhasan@iupui.edu

Baichuan Zhang

Facebook Inc
baichuan24@fb.com

Khalifeh AlJadda and Mohammed Korayem

CareerBuilder, Inc, Norcross, GA
(khalifeh.aljadda,mohammed.korayem)@careerbuilder.com

ABSTRACT

Job recommendation is an important task for the modern recruitment industry. An excellent job recommender system not only enables to recommend a higher paying job which is maximally aligned with the skill-set of the current job, but also suggests to acquire few additional skills which are required to assume the new position. In this work, we created three types of information networks from the historical job data: (i) job transition network, (ii) job-skill network, and (iii) skill co-occurrence network. We provide a representation learning model which can utilize the information from all three networks to jointly learn the representation of the jobs and skills in the shared k -dimensional latent space. In our experiments, we show that by jointly learning the representation for the jobs and skills, our model provides better recommendation for both jobs and skills. Additionally, we also show some case studies which validate our claims.

KEYWORDS

Job recommendation, Skill recommendation

ACM Reference Format:

Vachik S. Dave, Baichuan Zhang, Mohammad Al Hasan, and Khalifeh AlJadda and Mohammed Korayem. 2018. A combined representation learning approach for better job and skill recommendation. In *Proceedings of CIKM'18*. ACM, New York, NY, USA, Article 4, 9 pages. <https://doi.org/https://doi.org/10.1145/3132847.3132873>

1 INTRODUCTION

A well-planned and informed job transition is very important for a successful career, however, for many industries, making an effective and long-term job transition plan is difficult due to highly dynamic nature of job market in those industries. More often, an entirely new job sector emerges with unfamiliar job titles and unknown skill requirements. Also, skills needed to make a desired job transition change dramatically as new technologies emerge and old technologies become obsolete, which makes it very difficult to identify a collection of skills that are best associated with a job title.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CIKM'18, October 22–26, 2018, Turin, Italy.

© 2018 Copyright held by the owner/author(s).

ACM ISBN ISBN 978-1-4503-4918-5/17/11...\$15.00

<https://doi.org/https://doi.org/10.1145/3132847.3132873>

Curricula in an academic education also fall far behind to match the needs of many industries due to rapid technological and procedural evolution—as a result, schools fail to train an individual with the most up-to-date skill-set resulting skill gap [8].

To reduce or eliminate skill gap, accurate identification of skill mismatch between the skills expected by employers and those possessed by the job seekers is essential. This is not an easy task, as the same skill may have different level of utility depending on the job role. Hence, we need to compute a numeric score denoting the relevance of a given skill for a given job title. Mention frequency of a skill over different advertisements of a given job title is a poor indicator of relevance as it simply ranks the generic skills to the top. For instance, “Business Administration” is quite frequently mentioned as a needed skill for “Accountant” job, but it is not much informative or unique for that job title. Another important task in this regard is to compute the similarity among various skills, so that it is easier to identify which skills a candidate should acquire given his existing skill-set. Also the knowledge of skill similarity enables an employer to identify substitute skills when some needed skills are short in supply in the labor market. But, computing skill similarity is also not an easy task because such similarity values vary depending on the job role for which the skills are being considered. In summary, the intricacy of job titles and skills demands a principled approach for computing the relevance of a skill to a job title, and skill to skill similarity, so that they can be used in numerous downstream use cases in relation to job, and skill recommendation; obtaining such a solution is the focus of this work.

With the proliferation of online recruiting and job portals, such as linkedin.com, careerbuilder.com, indeed.com, and monster.com, machines learning, and information retrieval based models are being applied to analyze millions of resumes and job descriptions to provide service to both potential job-seekers and employers. For a job-seeker, these services include job recommendation by matching his professional profile with the description of thousands of available jobs. Specifically, all job portals match a candidate’s skills with the required skills mentioned in a job description to find a list of top jobs for which the candidate is most qualified. They also provide career advice in terms of skills that the candidate should acquire, and provide resume service by advising which skills should be highlighted in a resume. LinkedIn premier account provides a job-seeker a learning opportunity to develop highly sought out skills. Careerbuilder provides career-path guidance for

a job-seeker so that the candidate can better plan his or her next career move. Large enterprises consider skill re-purposing or skill fungibility for selecting candidates from the existing employees to train them to acquire skills which are short in supply in the market. A number of papers have been published which discuss how the above services are rendered by mining data from resumes and job descriptions [3, 4, 10, 11, 14].

However, the majority of the published works on job recommendation [11, 12] consider skill as a passive entity. Barring a few exceptions, skills are just a keyword associated with the job description, which need to be matched for finding the best job recommendation. For example, traditional recommendation system [11] based models consider user-job dyadic data for modeling recommendation using both content-based and collaborative filtering based methods. For these models, skill is not an active entity, they are used indirectly as keywords in job profile and user profile. Job recommendation has also been done by building generative models using sequences of job-history of many users [12]. Such a model only uses job history data ignoring skills associated with a given job. Even manually curated job ontology, such as O*NET (Occupational Information Network)¹, a program funded by US Department of Labor and Employment, lacks up-to-date information about the skills needed for performing a job. So, these works fail to capture the intricacy of job titles and skills, neither they provide a principled approach for computing the relevance of a skill to a job title, or skill to skill similarity.

In Careerbuilder Inc, we have invested enormously to understand the relationship between the skills and job titles. Specifically, we are interested to find a set of relevant skills in ranked order for each normalized job title. Along this effort, we have built an in-house term extraction system, called SKILL [19], that can extract skill keywords from both job advertisements and job seekers' resumes. The extraction system returns a confidence score for each extracted skill keyword, but this score represents the likelihood of the keyword to be a skill term, rather being the relevance of that skill term to a given job title. In a follow-up work [20], we have used Tf-Idf based score for computing a base relevant score of a skill for a job title; then we have modified this base score with different factors, including the uniqueness of a skill, and global and local frequency of the skill. This work solves the skill to job title relevance computation task, but it does not yield skill to skill similarity directly.

In this work, we present a representation learning framework for joint learning of representation vectors of both job titles and skills in a single vector space. The strength of the proposed framework is that it assimilate signals from multiple sources to yield latent vectors of job titles and skills so that a simple dot product between a job vector and a skill vector returns the relevance of that skill to that job, and likewise, a dot product between two skill vectors returns the similarity between two skills. The learned vectors can be used for better job recommendation and skill recommendation. The similarity values obtained from the latent vectors using our framework are more accurate because they consider intricate relationship between jobs and skills. Specifically, we use three networks

as input to this representation learning model, which are job transition graph, job skill graph, and skill co-occurrence graph and use both Bayesian personalized ranking and margin based loss functions to learn the vector representations. Experimental comparison with several baseline methods show that our proposed representation learning framework yields better representation vectors. The contribution of this work is summarized as below:

- We propose a representation learning based solution that leverages information of three graphs in order to represent each job and skill into a shared low-dimensional vector space for solving the job recommendation task. To the best of our knowledge, our work is the first one to adopt a representation learning framework for job recommendation.
- For the representation learning, we present two novel pairwise ranking based loss functions. Specifically, the first one is based on Bayesian personalized ranking and the second one is based on Margin ranking, which are particularly suitable for solving the job recommendation task.
- Extensive validations on the CareerBuilder dataset and real-world case studies demonstrate that our proposed embedding methodology consistently outperforms three state-of-the-art methods in terms of job recommendation task, which improves HR, NDCG, and pair-wise AUC by 3.4%, 6.7%, 1.2%, respectively.

2 PROBLEM FORMULATION

Notations: We first introduce notations used in this paper. Throughout the paper, scalars are denoted by lowercase alphabets (e.g., n). Vectors are represented by boldface lowercase letters (e.g., \mathbf{x}). The transpose of the vector \mathbf{x} is denoted by \mathbf{x}^T . Bold uppercase letter (e.g., \mathbf{X}) denotes a matrix, and the i^{th} column of a matrix \mathbf{X} is denoted as \mathbf{x}_i . The dot product of two vectors is denoted by $\langle \mathbf{a}, \mathbf{b} \rangle$. $\|\mathbf{X}\|_F$ is the Frobenius norm of matrix \mathbf{X} . Finally calligraphic uppercase letter (e.g., \mathcal{X}) is used to denote a set and $|\mathcal{X}|$ denotes the cardinality of the set \mathcal{X} .

The business goal of this work is to predict the future job of a given person from her current job position, so that this tool can be used for job recommendation. For solving this problem, we utilize job transition relations along with information on what skills are required for different jobs. Furthermore, we also obtain skill co-occurrence information from resume dataset. From this set of information, we generated three graphs, which are defined as below:

Definition 2.1 (Job transition graph). A job transition graph is represented as $G^{jj}(\mathcal{J}, \mathcal{E}^j, \mathcal{W}^j)$, where \mathcal{J} is a set of jobs, \mathcal{E}^j is a set of directed edges from one (source) job to other (target) job representing people with the former job changed to next job.

For jobs $u, v \in \mathcal{J}$, an edge from u to v is represented as $e_{uv} = (u, v) \in \mathcal{E}^j$. The total number of jobs is denoted as $n^j = |\mathcal{J}|$.

Definition 2.2 (Job skill graph). A set of skills are associated with different jobs in \mathcal{J} . This information is represented as bipartite graph $G^{js}(\mathcal{J} \cup \mathcal{S}, \mathcal{E}^{js})$, where \mathcal{J} is a set of jobs, \mathcal{S} is a set of skills with number of skills $n^s = |\mathcal{S}|$ and \mathcal{E}^{js} is a set of edges from a job to a skill representing the job requires the corresponding skill.

¹<https://www.onetonline.org/>

Definition 2.3 (Skill co-occurrence graph). Let $G^{ss}(\mathcal{S}, \mathcal{E}^s, \mathcal{W}^s)$ be an undirected skill co-occurrence graph, where \mathcal{S} is a set of skills, \mathcal{E}^s is a set of edges.

All three graphs contain useful information to predict a future job given a current job. Besides, they also contain information regarding skills that are associated to a given job. First, the job transitional graph shows that if a person with similar (to current job) job has changed to a specific target job, there is higher chance that the person with the current job also get the same target job in future. Second, the job skill graph and skill co-occurrence graph carry crucial information to find similarity between jobs, which helps to explore similar source and target jobs. To combine all the information from all three graphs, we design a novel network embedding approach that obtains latent representations for both jobs and skills, such that this latent job representation helps to predict the future job. Besides, the associated skills of the future job can be obtained by finding the nearest skill vectors of that job. Note that, job-skill graph already provides relevant skills of a given job, but embedding vectors of jobs and skills provide an opportunity to obtain a higher quality relevance score as these vectors are created by assimilating multiple information sources. An added benefit of embedding vectors are that they allow us to understand the relation of jobs and skills through visualization.

Formally, given job transition graph G^{jj} , job skill graph G^{js} and skill co-occurrence graph G^{ss} , our goal is to obtain k -dimensional representation of jobs (\mathbf{W}) and skills (\mathbf{W}') into a shared latent space. Here, $\mathbf{W} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_{n_j}^T]^T \in \mathbb{R}^{n_j \times k}$, where \mathbf{w}_i is i^{th} column of embedding matrix \mathbf{W} , which is the representation of the i^{th} job. Similarly, $\mathbf{W}' = [\mathbf{w}'_1^T, \mathbf{w}'_2^T, \dots, \mathbf{w}'_{n_s}^T]^T \in \mathbb{R}^{n_s \times k}$ is the skill representation matrix. The embedding matrices \mathbf{W} and \mathbf{W}' should preserve the connectivity information from graphs G^{jj} and G^{ss} , respectively. Additionally, these matrices also leverage signals from graphs G^{js} and through G^{js} , job similarity information propagates to \mathbf{W}' and skill similarity information propagates to \mathbf{W} .

3 METHODOLOGY

In this section, we discuss our proposed representation learning model. Our goal is to encode the local neighborhood structures captured by the three networks (see Definitions 2.1 2.2 2.3) into k -dimensional job and skill embedding matrices with strong job and skill recommendation abilities.

3.1 Model Design

First, we capture job transitional information from graph G^{jj} . The main intuition behind the proposed embedding model is that transition between similar jobs is more likely to happen than non-similar jobs. Therefore, neighboring nodes in the graph G^{jj} should have more similar vector representation in the embedding space than non-neighboring nodes. For example, for an edge $e_{xy} \in \mathcal{E}^j$ the vector representation \mathbf{w}_x of job x should be closer to \mathbf{w}_y compared to \mathbf{w}_z when $e_{xz} \notin \mathcal{E}^j$. We calculate the affinity score between two embedding vectors using a dot product operation, hence the affinity score between job x and job y is represented as $A_{xy}^j = \langle \mathbf{w}_x, \mathbf{w}_y \rangle$. More precisely, we are interested in having higher affinity score between job x and job y compared to job x and job z given $e_{xy} \in \mathcal{E}^j$

and $e_{xz} \notin \mathcal{E}^j$, i.e. $A_{xy}^j > A_{xz}^j$. We can model the probability function that preserves the order $A_{xy}^j > A_{xz}^j$ for given \mathbf{w}_x , \mathbf{w}_y and \mathbf{w}_z . Specifically, we utilize two functions for this modeling task, 1) sigmoid function $\sigma(v) = \frac{1}{1+e^{-v}}$ and 2) *ReLU* function $ReLU(v) = \max(0, v)$. In the following, we show the formulation only for sigmoid function, and it is very similar for the *ReLU* function. The probability that the order $A_{xy}^j > A_{xz}^j$ is preserved can be formulated as below:

$$P(A_{xy}^j > A_{xz}^j \mid \mathbf{w}_x, \mathbf{w}_y, \mathbf{w}_z) = \sigma(A_{xy}^j - A_{xz}^j) \quad (1)$$

where,

$$A_{xyz}^j = A_{xy}^j - A_{xz}^j = \langle \mathbf{w}_x, \mathbf{w}_y \rangle - \langle \mathbf{w}_x, \mathbf{w}_z \rangle \quad (2)$$

From equation 1, it is clear that the higher the value of A_{xyz}^j , the better the ordering is preserved. Hence, our goal is to maximize the probability for preserving all the ranking orders of all training triplets (x, y, z) , where $e_{xy} \in \mathcal{E}^j$ and $e_{xz} \notin \mathcal{E}^j$. We assume that all training triplets are sampled independently, thus the joint probability of preserving all training ranking orders, $P(> \mid \mathbf{W})$, can be represented as below:

$$\begin{aligned} P(> \mid \mathbf{W}) &= \prod_{(x,y,z) \in \mathcal{D}^{jj}} P(A_{xy}^j > A_{xz}^j \mid \mathbf{w}_x, \mathbf{w}_y, \mathbf{w}_z) \\ &= \prod_{(x,y,z) \in \mathcal{D}^{jj}} \sigma(A_{xyz}^j) \\ &= \prod_{(x,y,z) \in \mathcal{D}^{jj}} \sigma(A_{xy}^j - A_{xz}^j) \end{aligned} \quad (3)$$

Where \mathcal{D}^{jj} is a set of training triplets from graph G^{jj} . We aim to maximize the joint probability of training triplets (equation 3). For the computational simplicity, we minimize the negative log of this joint probability instead, which is shown as follows:

$$\begin{aligned} O^{jj} &= \min_{\mathbf{W}} - \ln P(> \mid \mathbf{W}) \\ &= \min_{\mathbf{W}} - \sum_{(x,y,z) \in \mathcal{D}^{jj}} \ln \sigma(A_{xy}^j - A_{xz}^j) \end{aligned} \quad (4)$$

The optimization objective shown in equation 4 helps to achieve desirable job embedding, where similar jobs have higher affinity score than non-similar jobs.

Similarly, we obtain the skill embedding matrix $\mathbf{W}' \in \mathbb{R}^{n_s \times k}$, which preserve the skill similarity information. For the skill co-occurrence graph G^{ss} , our goal is to obtain higher affinity scores for coexisting skills. This can be achieved using the following optimization objective:

$$\begin{aligned} O^{ss} &= \min_{\mathbf{W}'} - \ln P(> \mid \mathbf{W}') \\ &= \min_{\mathbf{W}'} - \sum_{(x,y,z) \in \mathcal{D}^{ss}} \ln \sigma(A_{xy}^s - A_{xz}^s) \\ &= \min_{\mathbf{W}'} - \sum_{(x,y,z) \in \mathcal{D}^{ss}} \ln \sigma(\langle \mathbf{w}'_x, \mathbf{w}'_y \rangle - \langle \mathbf{w}'_x, \mathbf{w}'_z \rangle) \end{aligned} \quad (5)$$

Where, \mathbf{w}'_i is the i^{th} column of the skill embedding matrix \mathbf{W}' , \mathcal{D}^{ss} is a set of training triplets sampled from graph G^{ss} and affinity score between skills x and y is denoted as $A_{xy}^s = \langle \mathbf{w}'_x, \mathbf{w}'_y \rangle$. The objective function shown in the equation 5 is to maximize the difference in

terms of affinity scores between coexisting skills and non-relevant skills in the graph G^{ss} .

Lastly, we incorporate information from job skill bipartite graph G^{js} into the job embedding matrix \mathbf{W} and skill embedding matrix \mathbf{W}' . In order to achieve that, we sample two skills y^s and z^s such that the skill y^s is associated with job x^j and the skill z^s is not relevant to job x^j . Here we compute the affinity score between job x^j and skill y^s as $A_{x^j y^s}^{js} = \langle \mathbf{w}_{x^j}, \mathbf{w}'_{y^s} \rangle$. Note that for a given job x^j , we select its corresponding job latent vector from \mathbf{W} and for the skill y^s we select its corresponding latent vector from \mathbf{W}' . The objective function is formulated as below:

$$\begin{aligned} O^{js} &= \min_{\mathbf{W}, \mathbf{W}'} -\ln P(> | \mathbf{W}, \mathbf{W}') \\ &= \min_{\mathbf{W}, \mathbf{W}'} - \sum_{(x^j, y^s, z^s) \in \mathcal{D}^{js}} \ln \sigma(A_{x^j y^s}^{js} - A_{x^j z^s}^{js}) \\ &= \min_{\mathbf{W}, \mathbf{W}'} - \sum_{(x^j, y^s, z^s) \in \mathcal{D}^{js}} \ln \sigma(\langle \mathbf{w}_{x^j}, \mathbf{w}'_{y^s} \rangle - \langle \mathbf{w}_{x^j}, \mathbf{w}'_{z^s} \rangle) \end{aligned} \quad (6)$$

Where \mathcal{D}^{js} is a set of training triplets sampled from graph G^{js} .

The goal of the proposed network embedding framework is to unify these three types of relations (G^{jj} , G^{js} , G^{ss}) together to learn high quality job and skill embedding matrices. An intuitive manner is to collectively embed these three graphs, which can be achieved by minimizing the following objective function:

$$\mathbf{O}(\mathbf{W}, \mathbf{W}') = \min_{\mathbf{W}, \mathbf{W}'} O^{jj} + O^{js} + O^{ss} + \lambda \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{W}'\|_F^2) \quad (7)$$

Where λ is a regularization co-parameter and $\|\cdot\|_F^2$ is l_2 regularization for both embedding matrices to avoid over-fitting.

We name the proposed method with sigmoid function as *Joint-BPR*, as equations 4, 5 and 6 are in the similar spirit to Bayesian Personalized Ranking (BPR) [13, 18]. Similarly, we name the proposed method with *ReLU* function as *Joint-Margin*.

3.2 Model Optimization

Our proposed embedding framework has two model parameters \mathbf{W} and \mathbf{W}' , which are job and skill embedding matrices. We learn these matrices using mini-batch gradient decent. Specifically, we sample triples (x, y, z) from each of the three graphs (G^{jj} , G^{js} , G^{ss}), where (x, y) are connected and (x, z) are dis-connected pair of nodes in the corresponding graphs as discussed in the previous section. For each mini-batch, we compute derivative of the objective function shown in equation 7 with respect to \mathbf{W} and \mathbf{W}' and update the matrix values using the following equations:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \alpha \times \frac{\partial \mathbf{O}(\mathbf{W}, \mathbf{W}')}{\partial \mathbf{W}} \quad (8)$$

$$\mathbf{W}'_{t+1} = \mathbf{W}'_t - \alpha \times \frac{\partial \mathbf{O}(\mathbf{W}, \mathbf{W}')}{\partial \mathbf{W}'} \quad (9)$$

where α is the learning rate. Additionally, we initialize both matrices from normal distribution with 0.0 mean and 0.1 standard deviation. For better understanding of the proposed methodology, we provide the pseudo-code of our method in Algorithm 1.

Algorithm 1: Pseudo-code for the proposed embedding framework

Input: G^{jj} , G^{js} , G^{ss} , embedding dimension k , batch size b , learning rate α , regularization coefficient λ
Output: Job embedding matrix \mathbf{W} and Skill embedding matrix \mathbf{W}' .

- 1: Initialize \mathbf{W}, \mathbf{W}' as k -dimensional matrices with 0 mean and 0.1 standard deviation from normal distribution
- 2: Given G^{jj}, G^{js}, G^{ss} , construct training triplet sets \mathcal{D}^{jj} , \mathcal{D}^{js} , and \mathcal{D}^{ss} respectively using uniform sampling technique
- 3: **for** each training instance in training sample sets **do**
- 4: Update involved parameters using min-batch gradient descent as described in Sections 3.1, 3.2
- 5: **end for**
- 6: **return** \mathbf{W}, \mathbf{W}' .

4 EXPERIMENTS AND RESULTS

We conduct comprehensive experiments to thoroughly study the proposed methods and compare them with the existing methods. Additionally, we perform experiments to study graph-wise contribution for job recommendation task and to analyze the sensitivity of model parameters.

4.1 Data Preparation

Our methodology relies on utilizing three networks, namely, (i) job transition graph, (ii) job-skill graph, and (iii) skill co-occurrence graph. In order to build these networks, we collected 20 million resumes from CareerBuilder.com, one of the largest human capital solution company in the US. We parse each resume to extract the work history section which is then parsed to extract the job title and employment date. We order the job titles in a resume in the temporal order such that job x is placed before job y if the employment date of x is earlier than the employment date of y . Therefore, in the job transition network an edge e_{xy} represents that job x listed in a resume before job y in the work history section. Lastly, we normalized the set of job titles using Carotene [9], an in-house job classification tool in CareerBuilder. This step reduces the number of unique jobs to 4325 with 2, 432, 231 distinct job transitions. Note that although job transition graph is directed, for learning the representation vector, we treat it as an undirected graph.

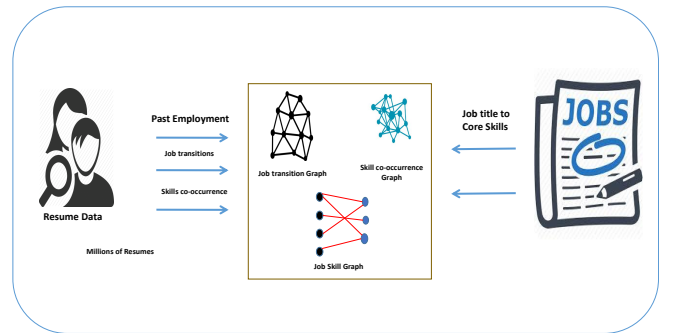


Figure 1: Data Preparation

Table 1: Comparison results for job transition recommendation. (Embedding dimension = 50)

Metrics	Bigram	AANE	PTE	<i>Joint-BPR</i>	<i>Joint-Margin</i>
HR@10	0.2742	0.8964	0.9260	0.9055	0.9575
NDCG@10	0.1479	0.7151	0.7332	0.7060	0.7826
local AUC	0.6682	0.9622	0.9722	0.9698	0.9835

For the skill co-occurrence network, we connect the skills extracted from each resume, so an edge $e_{s_1s_2}$ means both skills s_1 and s_2 are listed in the same resume at least once. We remove any edge with co-occurrence value smaller than 10 to avoid poor quality relations, that left us with 6214 unique skills co-occurring in 11,760,132 different ways. To further reduce the noisy connections, we calculate the weights on the edges using Point-wise Mutual Information (PMI) and use reasonable weight threshold to filter poor quality edges.

Lastly, to build the job-skill bipartite graph, we extracted each skill mentioned in the description of job x in all the job postings that has job x . We built job to skill connections by using another in-house method, described in [21], which creates a graph that connects 4325 jobs to 6214 skills using 103,073 edges.

4.2 Comparison Works

We conduct comparison experiments to show superiority of the proposed methods on our real-world dataset discussed in the previous section. For this experiment, we compare with three baseline methods.

- **Bigram**: It estimates the job transition probability based on first-order Markov assumption i.e. it uses popularity of the target job in the network to calculate the transition probability. It is considered as one of the most competitive method in practice for densely connected graphs.
- **AANE [7]**: Accelerated Attributed Network Embedding is a state-of-art embedding method that incorporates nodal attributes with topology using joint matrix factorization to learn low-dimensional network representation.
- **PTE [15]**: Predictive Text Embedding incorporates all three graphs, namely Job transition, Job skill and Skill co-occurrence, into single network representations using matrix factorization. The objective of this method is to minimize the distance between empirical similarity distribution and embedding similarity distribution using KL-divergence.

4.3 Experiment Settings

For conducting these experiments, we create train and test set using leave-one-out strategy i.e. for each job in G^{ij} we randomly keep one of the neighboring job as positive test instance and the remaining neighboring jobs as positive train instance. Ranking all jobs is a costly task, hence we uniformly select 100 jobs (over all jobs in the dataset) as negative test instances. For evaluation, we use Hit-Rate (HR@10) from top 10 ranked jobs and calculate Normalized Discounted Cumulative Gain (NDCG@10) up to position 10 for evaluation. We also calculate pair-wise/local AUC (Equation 10) for test set \mathcal{T} which includes pair (x, y) , where x is a test node and y

is a positive target job. We uniformly sample (100) negative jobs (TJ^{x-}) for each test node x .

$$\text{local AUC} = \frac{1}{|\mathcal{T}|} \sum_{(x,y) \in \mathcal{T}} \frac{1}{|TJ^{x-}|} \sum_{z \in TJ^{x-}} \mathbb{1}(A_{xy}^j > A_{xz}^j) \quad (10)$$

For the proposed method, we use mini-batch gradient descent for the optimization. Specifically, we set learning rate as 0.1, regularization coefficient value as 0.0001, batch-size as 100 and number of epochs as 20. We use the same configuration for PTE for fair comparison. Also, we keep the embedding dimension as 50 for all embedding methods (*Joint-BPR*, *Joint-Margin*, AANE and PTE). For AANE, we perform grid search to select regularization parameter $\lambda = 0.01$ from set $\{0.0001, 0.001, 0.01, 0.1\}$ and penalty parameter $\rho = 0.1$ from set $\{0.05, 0.1, 0.5, 1\}$.

4.4 Comparison Results

The comparison results are depicted in the Table 1. We observe that both proposed methods (*Joint-BPR* and *Joint-Margin*) outperform the Bigram method substantially. Additionally, the proposed methods also improve the performance over AANE; for example, *Joint-Margin* achieves around 7% increment in hit-rate and 10% increment in NDCG over AANE. Notably, PTE performs better than other baselines and also outperforms one of the proposed methods (*Joint-BPR*) by small margin. One of the possible reasons for good performance by PTE is that it uses all three graphs to learn the job embedding. However, our proposed method *Joint-Margin* outperforms all competing methods and improves the Hit-rate by 3.4% and NDCG by 6.74% compared to the second best method, which is PTE.

4.5 Job Clustering

We categorize all jobs into different groups using O*Net² job-categories. We plot our job embeddings generated from our proposed *Joint-Margin* to check the clustering behavior of the jobs into the latent space. As we have many different job-categories, we plot two similar categories into same figure to show that proposed embeddings can distinguish similar categories in the latent space, for example we plot "Business and Finance jobs" with "Sales and related jobs" in Figure 2a, "Healthcare Practitioners jobs" with "Personal care and service jobs" in Figure 2b and "Food preparation and serving jobs" with "Farming, fishing and forestry jobs" in Figure 2c. We also plot three non-similar groups together to show strong clustering behavior among jobs from the same job-category as depicted in Figure 3. Notice that, though we do not use these O*Net job-category information to generate job embedding, still

²<https://www.onetonline.org>

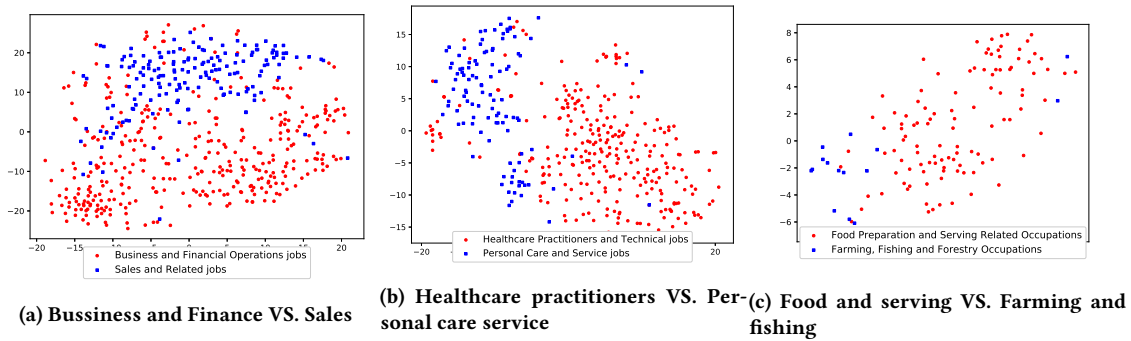
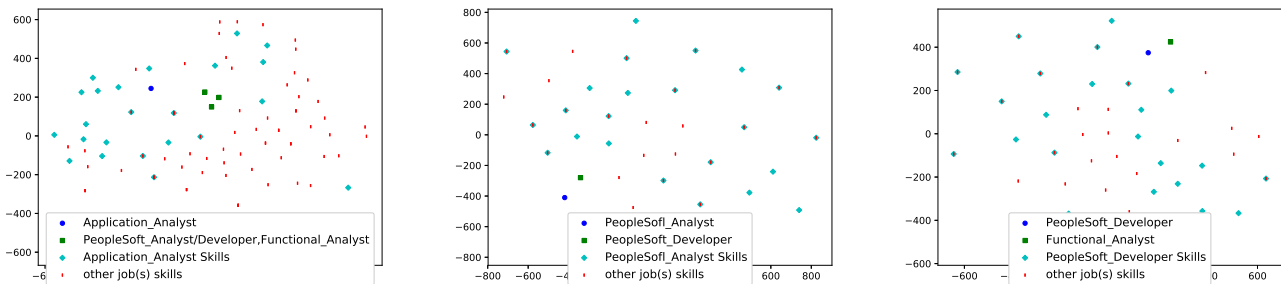


Figure 2: Clustering of similar job-categories in embedding space



Figure 3: Clustering of non similar job-categories in embedding space



(a) Distance between Application analyst and other jobs (b) Relation of PeopleSoft analyst and PeopleSoft developer (c) Relation of PeopleSoft developer and Functional analyst

Figure 4: Comparison of job transitions of a user

the proposed embedding generates correct clusters in the latent space.

4.6 Case Study

We study a professional, who switched 3 jobs during her career. She started working as "Application analyst" in August 1993 and switched to a new job as "PeopleSoft analyst" in January 1994. She worked there almost four years and then switched to a new job as "PeopleSoft developer" in October 1998. Finally after three years, in July 2001 she switched to a "Functional analyst" job and worked there until 2015. We observe that in the first job transition, the

professional switched the job very quickly, probably because the job domain was not very suitable for her. However, judging from her longer job duration in the later jobs, we suspect that those jobs were satisfying and the later job transitions were merely for better career prospect and career progression.

To check the quality of the proposed *Joint-Margin*, we check recommendation for these jobs, i.e., "Application analyst", "PeopleSoft analyst" and "PeopleSoft developer" using our job embedding. For "Application analyst" jobs, our recommendation suggests "PeopleSoft analyst" job at 45th position. On the other hand, for "PeopleSoft analyst" job, our method suggests "PeopleSoft developer" job at 4th

Table 2: Top 10 job and skill recommendations for 3 different job owners

Civil Engineer		Photographer		Cafe Manager	
Jobs	Skills	Jobs	Skills	Jobs	Skills
Structural Engineer	Civil Engineering	Creative Manager	Graphic Design	Operations Manager	Customer Service
Geotechnical Engineer	Surveying	Graphic Artist	Adobe InDesign	Food Service Manager	Management
Transportation Engineer	MicroStation	Production Artist	Printing	Restaurant Manager	Training
Traffic Engineer	Geotechnical Eng.	Art Director	Adobe Photoshop	General Manager	Leadership
Staff Engineer	Professional Eng.	Graphic Designer	Photography	Banquet Manager	Operations
Surveyor	AutoCAD	Production Director	Adobe Flash	Restaurant Supervisor	Rotation
CAD Technician	Land Development	Media Assistant	Adobe Dreamweaver	Store Director	Food Services
Transmission Engineer	Elevation	Package Designer	Adobe Creative Suite	Operations Manager	Retailing
Water and Wastewater Eng	Site Planning	Layout Designer (Arts)	Web Design	Field Service Manager	Sales
Architectural Designer	Physical Education	Videographer	Adobe Illustrator	Department Head (Sales)	Merchandising

rank. For the "PeopleSoft developer" job, the rank of "Functional analyst" job is 2. As we can see, the recommendation using the proposed embedding methodology provides good ranking for jobs that are satisfactory, and leads to better career prospect.

In Figure 4, we plot all four jobs and corresponding skills using job and skill embeddings to understand the recommendation behavior of our method. We observe from Figure 4a that "Application analyst" is far in latent space compared to other three jobs and skill set for "Application analyst" is also not well aligned with skill set of the other three jobs. However, there are a few overlapping skills which qualify "Application analyst" as a precursor for the other three jobs. Note that "Application analyst" is a generic job, from which 880 distinct job transitions happened in our dataset; among them "PeopleSoft analyst" is ranked 45 in recommendation, possibly because of the common skills between these two jobs. This explains the benefit of using skill information in the proposed embedding. In the remaining two plots in this figure, we show relation between "PeopleSoft analyst", "PeopleSoft developer" and "Functional analyst" jobs. As can be seen in Figures 4b, 4c, their skill sets are highly indistinguishable in the latent space as these jobs are well aligned in the career progression trajectory in that job sector.

4.7 Example of Job and Skill Recommendations

Here, we study recommendations for three job owners from different domains and show that the proposed method provides highly relevant suggestions. As we mentioned before, the proposed method embeds the jobs and skills into a shared latent space, thus we can provide job recommendation as well as skill recommendation. We list top 10 recommended jobs and skills for three different jobs in Table 2. From the table, we observe that the recommended jobs are typically more specialized job or a job that one can achieve after being promoted. For instance, a "photographer" is recommended "graphic artist" and "production artist" jobs, which are more specialized jobs than the photographer. He is also suggested jobs, such as "creative manager" and "art director", which are jobs that a photographer can obtain after a career advancement. For skill recommendation, we provide highly relevant (mandatory) and advanced skills required in the domain as shown in the Table 2. For example, for a good "photographer", knowing "photography", "printing" and "Adobe photoshop" are mandatory, but he could get suitable higher level jobs if he acquires skills such as "graphic

design", "web designing" and different Adobe tools. Similar observations can be made for the recommendation of other two jobs that we have studied.

Note that, by sharing information among three different input graphs our proposed model can recommend skills which have not been associated to a job in the job-skill bipartite graph. For instance, the graph G^{js} does not contain any connection between the job "civil engineer" and the skill "site planning". However, our proposed method is able to learn the relation between these two and put them nearer in the latent space such that the skill appears as top skills for the "civil engineer" job. This verifies our claim that while learning the job and skill representations, we leverage information from all three graphs and this method provides higher quality job and skill embedding.

4.8 Parameter Study

For the proposed method, there are two major influential parameters, first is the learning rate (α) and second is the embedding dimension (k). In this study, we analyze the influence of both parameters separately over the recommendation performance of the proposed methods. For both experiments, we keep the regularization coefficient, batch-size and epoch count the same as mentioned in Section 4.3.

Learning Rate Study

For this experiment, we keep the embedding dimensions as 50 and select the learning rate from set $\{0.01, 0.05, 0.1, 0.5\}$. We report the performance using Hit-Rate, NDCG and AUC in Figure 5. In the figure, we can observe that the performance of the proposed models improve drastically from learning rate 0.01 to 0.05, precisely 27.5% and 22.5% improvement in terms of NDCG@10 for *Joint-BPR* and *Joint-Margin* respectively. The possible explanation is due to the fact that with lower learning rate, the model converges slowly, while with higher learning rate, the model will converge to its optimum quickly.

Embedding Dimension Study

For this experiment, we keep the learning rate as 0.1 and select the embedding dimensions from the set $\{30, 40, 50, 60, 70\}$. We report the performance using Hit-Rate, NDCG and AUC in the Figure 6. This figure shows continuous improvement in the performance for larger embedding dimensions, however, this improvement is not noticeable after embedding dimensions reach 50. For example,

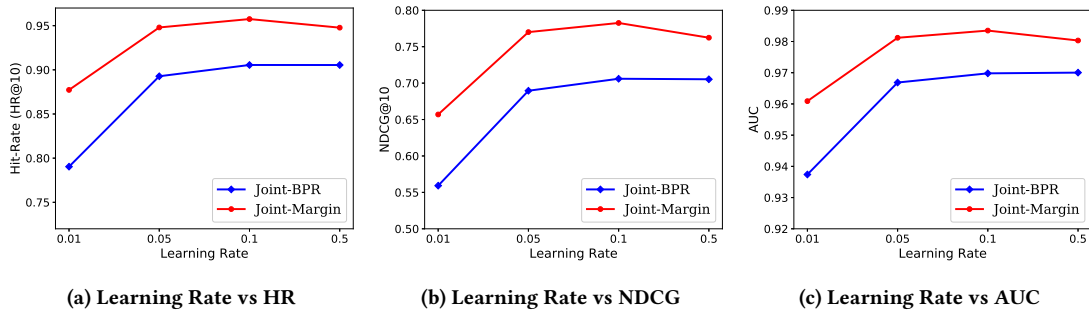


Figure 5: Performance of the proposed models for different learning rate values

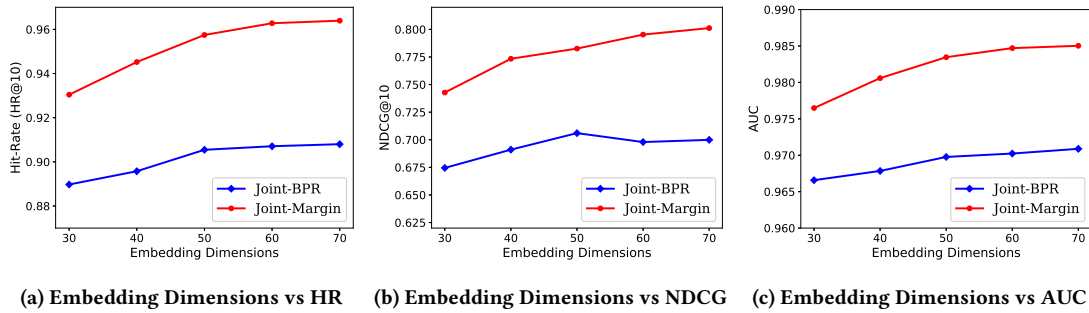


Figure 6: Performance of the proposed models for different embedding dimensions

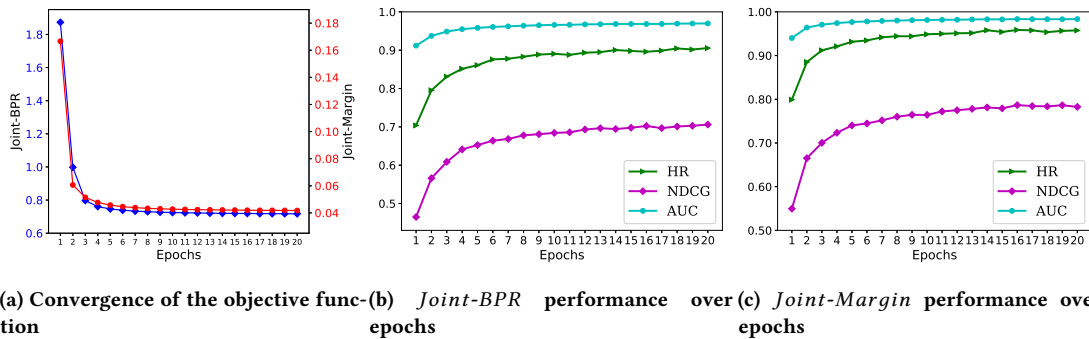


Figure 7: Convergence study for both proposed embedding methods

there is around 5% improvement in terms of NDCG@10 when the embedding dimension increases from 30 to 50 for *Joint-BPR*, but performance degrades in terms of NDCG@10 when the embedding dimension increases from 50 to 70.

4.9 Convergence Study

Finally we study the convergence behavior of the proposed methods. In particular, we calculate objective function value (Equation 7) at each epoch and plot the results in Figure 7a, where blue line represents *Joint-BPR* and red line shows the behavior of *Joint-Margin*. Figure 7a also shows that both models converge in 3-4 epochs. We also evaluate performance of the proposed models after each epoch using hit-rate, NDCG and AUC. For this evaluation,

results are depicted in Figures 7b and 7c. From both figures, we observe that the recommendation performance of both embedding models become steady after a few epochs.

5 RELATED WORK

In existing works on job recommendation, both collaborative filtering (CF) based methods and content-based filtering methods are used. CF methods use only historical transitions for providing recommendations [1, 11, 14]. Al-Otaibi et al. [1] provided detailed study on job recommendation methods using CF and also discussed challenges and limitations of CF such as sparsity, cold start, etc. For the content-based filtering, most of the works [2, 4, 6] use features (content) of the jobs and provide future recommendations using

feature level similarities. Guo et al. [5] discussed various content based and case based approaches for job recommendation with a comparative study. Diaby et al. [3] proposed a computational model that suggests jobs to different online social media users from their available profiles. In a recent work, the authors propose a hybrid approach [17] that leverages the strengths of both CF and content based filtering for job recommendation. Lu et al. [11] have built recommendation systems for job seeking and recruiting websites, in which traditional content based, relational based, and a hybrid model are used for computing a similarity value between a job and a candidate; then personalized multi-relational page rank is used for ranking the jobs in terms of preference of a candidate.

Relating to labor market, there exists a few other works which consider other tasks besides job recommendation. For example, the scientists in LinkedIn [16] used local sequence alignment for finding professionals whose career sequence matches with a given “ideal” career sequence. This tool can be used by recruiters to search the best candidate for a given job. Mimno et al. [12] proposed a generative model where the authors obtain a set of topics from the job description and then build a generative model for a sequence of jobs in a resume. This model can be used for building career trajectory. Faizan et al. [9] proposed models for job classification system, and Zhao et al. [19] proposed methods for skill extraction and normalization.

Latent embedding based method has not received much attention for the job market domain except for the work of Li et al. [10], which provides an embedding based approach to predict next career move of a user. It utilizes signals from two sources, profile context matching and career path mining, with a contextual LSTM model for predicting the next step in career. However, the objective of our work is different from Li et al. and our proposed embedding learning method is also very different. Specifically, we use latent embedding based approach that combines the job and skill information and learns the representation of both jobs and skills in a shared latent space.

6 CONCLUSION

To conclude, in this paper we propose a novel representation learning based solution to address the job and skill recommendation task. Our proposed representation learning model utilizes the pairwise ranking objective which learns job and skill vector representations into a shared latent space using three pre-processed graphs. This joint embedding approach not only allows us to provide high quality job recommendation but also provides skill suggestions required to obtain the new job. Our experimental results on the CareerBuilder dataset and case studies demonstrate that our proposed methodology consistently outperforms several existing state-of-the-arts for the job and skill recommendation.

A limitation of our proposed representation learning framework is that it is transductive, i.e., it learns representation vectors of jobs and skills that are available in the input graphs. But, in CareerBuilder, we often observe new job titles and skills, and our model is needed to be retrained to obtain representation vectors of these entities so that we can utilize them in the job and skill suggestion.

An inductive learning framework is needed to overcome this limitation. We leave the development of such a framework as our future work.

REFERENCES

- [1] Shaha T Al-Otaibi and Mourad Ykhlef. 2012. A survey of job recommender systems. *International Journal of Physical Sciences* 7, 29 (2012), 5127–5142.
- [2] Nikolaos D Almalis, George A Tsihrintzis, Nikolaos Karagiannis, and Aggeliki D Strati. 2015. FoDRA: A new content-based job recommendation algorithm for job seeking and recruiting. In *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on*. IEEE, 1–7.
- [3] Mamadou Diaby, Emmanuel Viennet, and Tristan Launay. 2013. Toward the next generation of recruitment tools: an online social network-based job recommender system. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 821–828.
- [4] Shiqiang Guo, Folami Alamudun, and Tracy Hammond. 2016. RésumMatcher: A personalized résumé-job matching system. *Expert Systems with Applications* 60 (2016), 169–182.
- [5] Xingsheng Guo, Houssein Jerbi, and Michael P O’Mahony. 2014. An analysis framework for content-based job recommendation. In *22nd International Conference on Case-Based Reasoning (ICCBR), Cork, Ireland, 29 September–01 October 2014*.
- [6] A. Gupta and D. Garg. 2014. Applying data mining techniques in job recommender system for considering candidate job preferences. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 1458–1465.
- [7] Xiao Huang, Jundong Li, and Xia Hu. 2017. Accelerated Attributed Network Embedding. In *Proceedings of the SIAM International Conference on Data Mining*. 633–641.
- [8] Faizan Javed, Phuong Hoang, Thomas Mahoney, and Matt McNair. 2017. Large-Scale Occupational Skills Normalization for Online Recruitment.
- [9] F. Javed, Q. Luo, M. McNair, F. Jacob, M. Zhao, and T. S. Kang. 2015. Carotene: A Job Title Classification System for the Online Recruitment Domain. In *2015 IEEE First International Conference on Big Data Computing Service and Applications*. 286–293. <https://doi.org/10.1109/BigDataService.2015.61>
- [10] 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*. International World Wide Web Conferences Steering Committee, 505–513.
- [11] Yao Lu, Sandy El Helou, and Denis Gillet. 2013. A recommender system for job seeking and recruiting website. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 963–966.
- [12] David Mimno and Andrew McCallum. 2008. Modeling career path trajectories.
- [13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.
- [14] Walid Shalaby, BahaaEddin AlAila, Mohammed Korayem, Layla Pournajaf, Khalifeh AlJadda, Shannon Quinn, and Wlodek Zadrozny. 2018. Help Me Find a Job: A Graph-based Approach for Job Recommendation at Scale. *arXiv preprint arXiv:1801.00377* (2018).
- [15] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding Through Large-scale Heterogeneous Text Networks. In *SIGKDD*. 1165–1174.
- [16] 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 (KDD '14)*. 1945–1954.
- [17] Shuo Yang, Mohammed Korayem, Khalifeh AlJadda, Trey Grainger, and Sriraam Natarajan. 2017. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowledge-Based Systems* 136 (2017), 37–45.
- [18] Baichuan Zhang and Mohammad Al Hasan. 2017. Name Disambiguation in Anonymized Graphs Using Network Embedding. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 1239–1248.
- [19] Meng Zhao, Faizan Javed, Feroosh Jacob, and Matt McNair. 2015. SKILL: A System for Skill Identification and Normalization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 4012–4017. <http://dl.acm.org/citation.cfm?id=2888116.2888273>
- [20] W. Zhou, Y. Zhu, F. Javed, M. Rahman, J. Balaji, and M. McNair. 2016. Quantifying skill relevance to job titles. In *2016 IEEE International Conference on Big Data (Big Data)*. 1532–1541.
- [21] W. Zhou, Y. Zhu, F. Javed, M. Rahman, J. Balaji, and M. McNair. 2016. Quantifying skill relevance to job titles. In *2016 IEEE International Conference on Big Data (Big Data)*. 1532–1541.