JobMiner: A Real-time System for Mining Job-related Patterns from Social Media

Yu Cheng, Yusheng Xie, Zhengzhang Chen, Ankit Agrawal, Alok Choudhary
EECS Department, Northwestern University
{ych133,yxi389,zzc472,choudhar}@eecs.northwestern.edu

Songtao Guo
LinkedIn
soguo@linkedin.com

ABSTRACT

The various kinds of booming social media not only provide a platform where people can communicate with each other, but also spread useful domain information, such as career and job market information. For example, LinkedIn publishes a large amount of messages either about people who want to seek jobs or companies who want to recruit new members. By collecting information, we can have a better understanding of the job market and provide insights to job-seekers, companies and even decision makers. In this paper, we analyze the job information from the social network point of view. We first collect the job-related information from various social media sources. Then we construct an inter-company job-hopping network, with the vertices denoting companies and the edges denoting flow of personnel between companies. We subsequently employ graph mining techniques to mine influential companies and related company groups based on the job-hopping network model. Demonstration on LinkedIn data shows that our system JobMiner can provide a better understanding of the dynamic processes and a more accurate identification of important entities in the job market.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Data Applications—*Data Mining*

General Terms

Design, Experimentation, Human Factors

Keywords

Social media, temporal network, influence analysis, graph mining, job market

1. INTRODUCTION

Social media has become one of the most popular web and mobile applications, which provides a platform for Internet users sharing messages and communicating with others. According to a study, hundreds of millions of Internet users spend about 20% to 25% of their time on social media websites like Facebook, Twitter, Youtube and Flickr in a day [3]. Online social media platforms also play a critical role in spreading useful domain information such as job-related messages either about newly employed people or new job opening. For example, companies actively maintain pages on Twitter to interact with online users and post a lot of tweets about new job openings like "Looking for an exciting new Graphics or Animation job? We are hiring!" by Adobe. A more typical social media platform is LinkedIn, the largest professional social network. Besides that, we can often see personnel flow information from Twitter like: "John Smith joined Amazon as a software engineer, previously in Walmart" or "Rebecca Wheeler joined Goldman Sachs as a vice president, previously in Accenture". Figure 1 shows an example of a list of posted messages about job opening information on Twitter and LinkedIn, respectively.

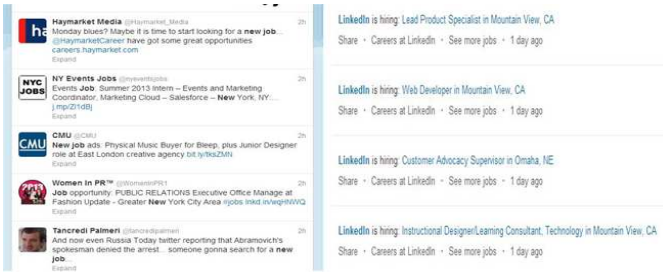

Figure 1: Examples from Twitter and LinkedIn about job market information.

Laid with such rich information about job market, social media provides us with a new way to view and analyze it, driven by commercial marketing interests and online users' needs. How to effectively and efficiently discover job-related patterns from social media data is of great importance to increase product-marketing performance. Moreover, with the help of knowledge discovery techniques, we could detect the latent company activities and user behaviors on the job market instead of merely summarizing the statistics.

We propose a system called JobMiner to mine the job-related patterns of both individuals and companies on the job market using social media data. With the help of API, two kinds of information are collected from Twitter, Linkedin and Facebook: (1) job opening information by companies in real-time and (2) person's job-hopping information among

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

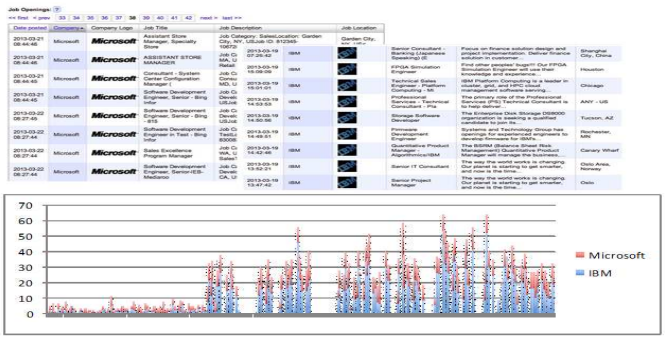

Figure 3: Monitored job openings of two companies.

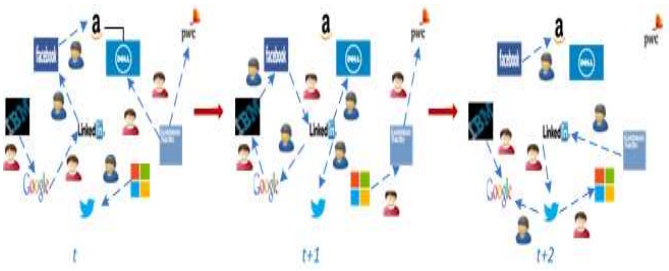

Figure 4: Job-hopping network for three time stamps.

companies. We monitor these information and detect related company groups and rank influential companies based on job-hopping network model. The major contributions of this work are: (a) we introduce a method to detect the job-related patterns from social media data; (b) we propose a network based modeling of job market; (c) we present data mining algorithms to estimate representativeness and company influences and to find related company groups; and (d) we implement a working system to demonstrate the effectiveness of the proposed approaches.

2. SYSTEM FRAMEWORK

As shown in Figure 2, JobMiner works as follows: (1) download job related content from social media; (2) generate a temporal network model for user job-hopping behavior; (3) perform graph mining based on the job-hopping temporal networks; and (4) provide the web interface for browsing, category-based company ranking and interest-based recommendation, such as recommending jobs or companies to users who may be interested in them.

2.1 Data Collection

For LinkedIn data, we use the stream API to download all possible job-related information without violating users' privacy. For example, for each LinkedIn company wall, we keep all public comments, "likes", and public user profiles.

2.2 Monitoring Job Openings

JobMiner monitors multiple channels of social media. These channels include over 10,000 LinkedIn company profiles. For most of these sources, we have all the posted job and personal job changing information to date since 2009 or its inception. These information is generally organized and shown in time order. We keep our data synchronized with the source media in real time. Figure 3 shows an example of posted job information for two specific companies: Microsoft and IBM, from April 2012 to March 2013. The red points represent the number of jobs posted by Microsoft while the blue ones represent the number of jobs posted by IBM. The JobMiner system allows users to keywords-search for job openings including job titles such as "software engineering" or job locations such as "Great Chicago Area".

2.3 Modeling Job-hopping Activities

To model the job-hopping activities between different companies, we employ a weighted directed graph $G = (V, E, t)$, where V is the set of nodes representing different organizations or companies, and E is the set of edges, representing personnel flows between pairs of different companies. The weight of edge $e_{u,v}$ represents the number of job-hopping activities from company node u to company node v.

Figure 4 shows an example of such network within three time stamps. The directions of the arrows represent the personnel flow directions. For example, there are people switch jobs from IBM to Google or from LinkedIn to Facebook at time stamp t. Generally, the time stamp interval should be set by balancing both network density and real time performance. For simplicity, we currently set the time stamp window as 1 month in our system.

2.4 Ranking Inf uential Companies

The activities of some highly influential companies would affect the other related companies on the job market. For example, if one company has some new job openings with high salaries, the employees at its peer companies might consider switching jobs. The task of ranking the influential companies on the job market is based on the generated job-hopping network model. The results can be used for recommending top-ranked companies to the job-seekers who are interested in them.

We mine such knowledge from two different viewpoints. First, given the posted jobs information, we can rank the companies in the given company category or the jobs of the corresponding topic, e.g., the system would rank the companies based on the number of posted jobs as "software engineer". Second, we generate the directed job-hopping graphs and identify influential entities by mining the job-hopping temporal network. Identifying influential entities has become an essential part of analyzing and understanding networked systems with application to a wide range of fields and applications. In this study, we assume that the time during which a network is observed is finite. We measure the influential of nodes based on: temporal degree, temporal closeness and temporal betweenness [5]. User can rank the influence of companies based on different measurements. We can further use the result to find a small set of influence entities in the network on a given topic based on influence maximization [6].

2.5 Discovering Related Companies

Given a company, we define its "related companies" as a group of companies that have dense connections in its generated job-hopping network. A simple approach to find these companies is to detect temporal communities [2] in the networks. We introduce a method that aims to find a good partition in a given network snapshot by exploiting the knowl-

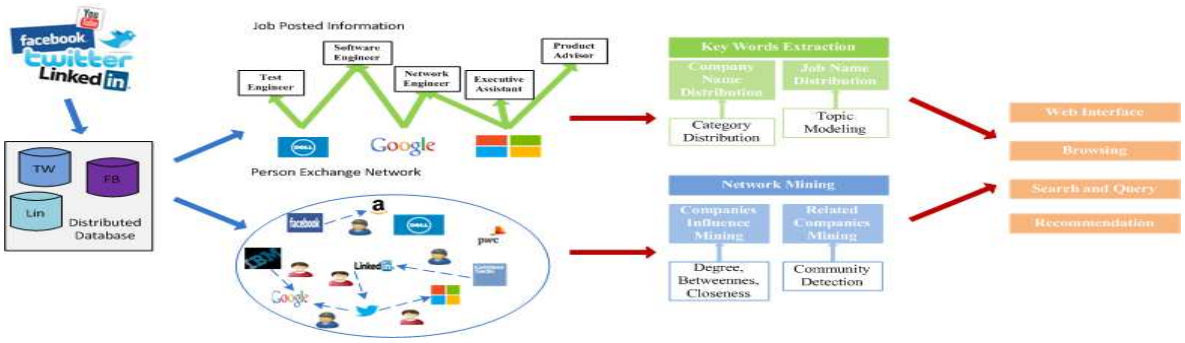

Figure 2: The system architecture for jobMiner.

edge of the community structure in the previous snapshot. The rationale behind the method is that any system tends to maintain some temporal contiguity in its features as it evolves.

More formally, given network snapshots G_{t-1}, G_t and the partition P_t with respect to t, the problem is to find a partition that solves the following constrained optimization problem:

$$\begin{aligned} &\text{maxmize} && \text{Q(P)} \\ &\text{s.t.} && \text{E(P)} < \zeta \end{aligned} \tag{1}$$

where Q is a quality function for the community structure in a snapshot, P denotes the space of all partitions, and E is a measure of distance or dissimilarity between the community structure at times t and $t-1$. The formulation above is based on the intuition that temporal communities can be detected by optimizing for quality in the current snapshot while ensuring that the distance from the past community structure is limited to a certain amount, as specified by the parameter ζ, where the parameter ζ is between 0 and 1. We set ζ as 0.5 in our experiments. To measure Q, we use modularity [7], a widely studied and tested quality function, which is defined as follows:

$$Q = \frac{1}{2M}\sum_{uv}[w_{uv} - \frac{k_u k_v}{2M}]\delta(C_u, C_v) \tag{2}$$

where w_{uv} is the weight of $e_{u,v}$, M is the total number of edge weighs, C_u is community which node u belongs to, k_u is the total degree of node u, and $\delta(i,j)$ is 1 if and only if $i = j$, and 0 otherwise.

To measure partition distance E, we first introduce the definition of E. Given network snapshots G_{t-1}, G_t and partition P_{t-1}, P_t, an edge $e_{(u,v)}$ in G_t is said to be estranged if edge $l_u \neq l_v$, given that u and v were neighbors in G_{t-1} and $l_u = l_v$ in P_{t-1}. E is now defined as the fraction of estranged edges in G_t and can be written as:

$$E = \frac{\sum_{u,v \in G_t} Z_{uv}(1 - \delta(l_u^t, l_v^t))}{2M} \tag{3}$$

where $Z_{uv} = \delta(l_u^{t-1}, l_v^{t-1})\sqrt{A_{t-1}A_t}$, A_{t-1}, A_t are the adjacency matrices of G_{t-1}, G_t, respectively.

Greedy local optimization methods used for modularity maximization cannot be directly used to solve the constrained optimization problem [8, 10]. Once this method of computing the Lagrange dual has been determined, we solve the dual problem of finding the best Lagrange multiplier by using Brent's method [1] and adapting a hierarchical version of the Label Propagation Algorithm [9].

Figure 5: The main page of JobMiner system.

Table 1: The details of the LinkedIn data.

Job Opening Data		
#posted jobs	#companies	#job categories
15,034,002	9879	342
Job Hopping Data		
#job hopping	#companies	#unique users
12,854,786	10381	12,854,779

3. BIG DATA SOLUTION

The JobMiner system collects large amount of data from social media. For example, in LinkedIn, there are nearly 4,000,000 posted jobs and 3,500,000 job-hopping activities from 10,000 companies from June 2012 to Dec. 2012. Such big data presents a number of challenges due to its complexity. One key challenge is how we can store the data, and how we can analyze and understand it given its size and our computational capacity. JobMiner is designed to be a data mining system that is (almost) always available, highly durable, and easily scalable for big data. JobMiner contains a cluster of several transactional databases and high-dimensional data warehouses. Each social channel of JobMiner (e.g., LinkedIn) updates in real time. More importantly, JobMiner can discover knowledge from cross-platform heterogeneous information and therefore give end-users timely and unbiased insights. To meet the needs of data mining tasks [4], we enable free text search in our system through multiple scalable indexing /searching solutions.

Figure 6: Top ranked companies in the management consulting category.

Figure 7: The related companies information of IBM.

4. DEMONSTRATION

We collect data from LinkedIn (http://www.linkedin.com/) from March 2011 to March 2013 to demonstrate the usage of JobMiner. Table 1 shows the details of the used data which consists of two parts: job opening data and job hopping data. Figure 5 shows the home page of our JobMiner system. The website is supported by Apache and MySQL and the user interface is supported by PHP and HTML. Some packages like Gephi (https://gephi.org/) and YUI (http://yuilibrary.com/) are used for data visualization. The user can log into the system from the main page. This page allows users to make search and/or query operations with specific keywords, such as a company category "Information Technology and Services", a job location "San Francisco Bay Area", or a job title "Systems Architect". JobMiner also provides information about the top ranked companies in each corresponding category. Figure 6 shows the top ranked companies returned by JobMiner system when the user uses "management consulting" as the keyword to search for jobs. Then the user is likely to check these companies to see if there are any suitable jobs available. Thus, the user chooses one company that he/she is interested in, like IBM and wants to find a job there. Unfortunately, JobMiner indicates that there is no job opening at IBM at that time. The system guides him/her to the LinkedIn activity page and finds him/her some other related companies on the job market as shown in Figure 7. The user could find other job opportunities with the help of this useful information. As mentioned in subsection 2.2, the JobMiner system helps users to monitor the posted jobs in real-time and provides a way for users to search jobs in the data stream.

5. CONCLUSION

In this paper, we have demonstrated a system called JobMiner to mine the latent behavior and relationships of people and companies on the job market using social media data. Possible application scenarios of JobMiner could be job recommendation system for both job-seekers and employers, or an auxiliary tool to analyze the job market information as well as its societal impact.

Note that this demo has used LinkedIn data as an illustrative example since it is currently the most professional social media websites for job hunting. However, the technique proposed can be applied/extended to other job-related social media data, such as Twitter and Facebook, or we can combine all these information together to form a hybrid job-related source system, which we intend to pursue in the future.

6. REFERENCES

[1] R. P. Brent. *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Inc., Englewood Cliffs, New Jersery, 1973.

[2] Z. Chen, W. Hendrix, and N. F. Samatova. Community-based anomaly detection in evolutionary networks. *J. Intell. Inf. Syst.*, 39(1):59–85, 2012.

[3] Y. Cheng, Y. Xie, K. Zhang, A. Agrawal, and A. Choudhary. Cluchunk: clustering large scale user-generated content incorporating chunklet information. In *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, BigMine '12, pages 12–19, 2012.

[4] Y. Cheng, K. Zhang, Y. Xie, A. Agrawal, W.-k. Liao, and A. Choudhary. Learning to group web text incorporating prior information. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 212–219. IEEE Computer Society, 2011.

[5] H. Kim and R. Anderson. Temporal node centrality in complex networks. *Physical Review E*, 85:026107+, Feb. 2012.

[6] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 199–208, New York, NY, USA, 2010. ACM.

[7] M. E. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci U S A*, 103(23):8577–8582, June 2006.

[8] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106+, Sept. 2007.

[9] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. Detecting communities and their evolutions in dynamic social networks–a bayesian approach. *Mach. Learn.*, 82(2):157–189, Feb. 2011.

[10] Y. Yang, J. Wang, and A. E. Motter. Network observability transitions. *Phys. Rev. Lett.*, 109:258701, Dec 2012.