A Comprehensive Recommender System for Fresher and Employer

Bhavna Gupta[1], Sarthak Kanodia[1], Nikita Khanna[1], Saksham [1],

[1]Department of Computer Science, Keshav Mahavidyalaya,
University of Delhi
{guptac7@gmail.com, sarthakkanodia@hotmail.com, nikita_khanna95@outlook.in,
khrn.saksham001@gmail.com}

Abstract. Due to overwhelming data on social networking sites about jobs and candidates, it becomes a time consuming task to generate a match between candidates and employers. Moreover, recruitment of a candidate, who has no work experience called as fresher, poses a two way problem. Firstly, the candidate due to a lack of experience is not able to decide upon a job among various opportunities which could utilize his/her maximum potential whereas the employer does not get any past referrals for the candidate to help in the process of recruitment. The proposed study addresses this problem by assisting both; a fresher with a recommended list of job openings which could interest him/her and the employer with a recommendation list of freshers which can be relied upon for the job. The study is assessed and validated with a series of experiments using real data from a social networking site, LinkedIN.

Keywords: Attributes, Similarity, Ratings, Recommender System

1 Introduction

Social networking sites (LinkedIN) provide a platform to connect job seeker(s) and employer(s). While job seeker has a perfect job in his mind, employer also has a picture of an ideal candidate for its job. Their goals are difficult to achieve if the job seeker (fresher/student) has little or no work experience, as work experience gives benefits to both parties: job seekers as well as recruiters. It lays future path for job seekers to make them reach to their destination whereas it generates referrals for recruiters with the help of which they can rely on the credibility of the candidate and minimize risk on their resources. To minimize the affect of work experience on students, a lot of effort and time is devoted both on-campus and off-campus, such as internship opportunities are provided to them from campus and students are provided with help and suggestion from their friends/seniors and family. In addition, job fairs and placement drives are held regularly to channelize students and employers to understand each other's requirements.

It is often realized that different sources (social networking sites, job fairs, placement drives) generate huge amount of data for the fresher, about number of job descriptions and for the employer, about candidate profiles. So there is a need for an

efficient mechanism, for the freshers and as well as employers, which can filter useful information from this huge amount of data.

The paper proposes a two-phase system, which serves both, the fresher and the employer, by providing each of them a recommendation list meeting their needs. For recommending employers to a fresher, similarity between the fresher Meta data (skill set, internship program), and the graduates who are in the job is obtained. Using similarity measure and applying threshold according to the personalized choice of fresher, a list of employers is generated. This list is further refined on the basis of employer ratings. This accumulated list of top k employers is provided as recommendation list to fresher. To get the recommendation list of potential freshers for an employer, the data of their previously employed recent students is taken and similarity is computed between them and freshers who have applied for the job. This similarity measure helps to generate a recommendation list of all those potential freshers for the employer with which their company will get benefited.

The rest of the paper is organized as follows. Section 2 discusses the related work of and section 3 details our proposed work. The various experimental results are elaborated in section 4 followed by conclusion and future work in section 5.

2 Related Work

Recommender system helps in filtering huge amounts of data/information while making a decision. But unlike traditional recommender system, job recommender system recommend one type of user (e.g. Job Applicant) to another type of user (e.g. Employer) [3][6]. Job recommender system uses different approaches that are presented and discussed in [11]. [9] proposed to use standard parameters which holds distinctive values for a job seekers while [1][8] have improved the results of job recommendation by providing weights to both job seekers' and jobs' different fields. Semantic and tree based knowledge matching process is discussed to get a profile similarity score with jobs in [13]. It is also been reflected how different profile patterns, similarity patterns and users' interactions can improve the results altogether [10]. Job recommendation framework based on rating of employers and job seekers' nearest neighbor is discussed in [7]. [4] used basic & knowledge attributes for employment and various psychological and social relationship attributes to improve the similarity score between job seekers and employers. Both parties (job seekers, employers) needs is termed as Reciprocal approach is addressed in [2]. [5] used hybrid approach of content & collaborative filtering techniques. In [12] content-based and interaction-based relations are translated into edges connecting different entities (candidates, employers and jobs).

The paper is focused specifically for generating a match between freshers (who do not have any job experience) and employers. Work experience helps both candidates and employers as it reduces the risks on employers' resources as well as for candidates it guides in laying the future path.

3 The Proposed System

The currently running recruiting systems are facing problem due to overloading of information from both the parties: job seekers and employers. This problem increases multifold, if job seeker is fresher i.e. has no prior work experience which makes the employer helpless as he/she has no details of past experience of candidate to get the referrals and lack of experience makes the job seeker confused to decide among the job opportunities. To address this problem, a comprehensive job recommender system is proposed in this paper, which recommends (1) employers to the aspiring freshers (2) eligible freshers for the specified job to the employers.

The system is built in two phases which is also represented diagrammatically in figure 1.Various components involved in the system are described in the following subsection.

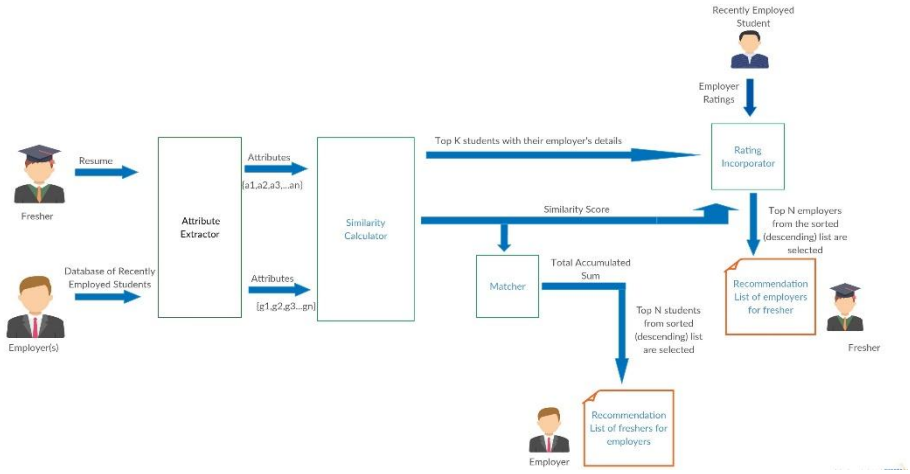

Fig 1: Job Recommender System for Fresher and Employer

3.1 Attribute Extractor

This component is supplied with either a fresher's profile or profile of the recently employed students at the employer depending upon the phase in which system is getting used. This module extracts relevant attributes from the input to generate a list of relevant attributes that will be further utilized further. The attribute extractor module makes sure that relevant attributes should be selected.

To extract attributes from profile of the fresher, the attributes are defined in two categories, i.e., bool and discrete. Fresher's information such as, Internship Company, internship title, institute and course are considered as bool whereas Internship duration is considered as discrete attribute. Each fresher's profile is

formally defined as u = {a_1, a_2, a_3, …. , a_n} where u is the student and a_1, a_2,….a_n are the fresher's attributes as extracted by the module.

In second phase, the database of recently employed students with recruiter is supplied as input to module. The output for this module will be e = {g_{i1}, g_{i2}, g_{i3}… g_{in}}; where e is the employer and g_{i1}, g_{i2}, g_{i3}… g_{in} are the attributes of i^{th} employed student.

3.2 Similarity Calculator

In first phase, this component computes the similarity between fresher and recently employed students at various employers. Similarity is calculated for both discrete and bool attributes.

For bool attributes, similarity is computed using equation 1.

$$S(u,g) = \sum_{i=1}^{n} w_i * sim(u_{a1}, g_{a1i}) \quad (1)$$

and

$$sim(u_{a1}, g_{a1i}) = \begin{cases} 1 & u_{a1} = g_{a1i} \\ 0 & u_{a1} \neq g_{a1i} \end{cases}$$

Whereas for discrete attributes, the similarity is computed using equation 2

$$S(u,g) = \sum_{i=1}^{n} wi * (1 - \frac{(|u_{a1} - g_{a1i}|)}{a_{imax} - a_{imin}}) \quad (2)$$

Where S is the job applicant/fresher looking for job; u_{a1} is a_1 attribute of the fresher; g_{a1i} is a_1 attribute of i^{th} recently employed student by the employer; w_i is the adjustable weight assigned to attribute; a_{imax}, a_{imin} are the maximum, minimum values of the i^{th} attribute whose similarity is being compared .

3.3 Rating Incorporator

This module works for phase 1. The input of this module is the detailed list of top k recently employed students with their similarity scores above a threshold with fresher and ratings of their employers as given by whole of their staff. This collective rating of particular employer is taken as his/her reputation. The similarity score and the ratings are combined using appropriate weights for each selected employer as shown in equation 3 and final recommendation list is prepared.

$$sim_{acc} = w1 * S(u, g_i) + w2 * Repu(e_j) \quad (3)$$

Where w1 and w2 are weights assigned to similarity score and ratings of j^{th} employer of i^{th} student present in the list prepared similarity module.

3.4 Matcher

For the second phase, after calculating the similarity between current fresher/job seeker and the recently employed student, a matching between a given fresher u and a potential employer e is need to be defined. This is the function of Matcher module. An employer will be defined as a set of similarity score of its recently employed students as e= $\{g_1, g_2, g_3, \ldots.., g_n\}$ with the fresher; where e is the employer and g is its recently employed students. The matching is defined as in equation 3.

$$M(u_j, e) = \sum_{i=1}^{m} S(u, g_i) \tag{3}$$

where S (u, g_i) is the similarity of fresher with the i^{th} recently employed students of the employer. The M (u_j,e) is computed for each j^{th} fresher and placed in a list. The top matching freshers from obtained sorted list is the recommendation list for the employer.

Summarizing, in first phase, the similarity between u (the fresher/ job seeker) and g (recently employed student who has offer and who will also provide employer rating) is computed by similarity calculator module and placed in set G. This similarity calculation is based on various bool and discrete attributes. The recently employed students in G set are then sorted and arranged according to their similarity weights. Then, we select the top K students from set G and also obtain their employers. Once the list is obtained then collective rating of each employer present in the list is taken and combined with its similarity score in rating incorporator module. Finally the sorting is done based on this average rating in descending order and the top N employers are recommended to the fresher u.

In second phase, the similarity between the freshers seeking job and employer' employed students are computed using similarity calculator. These similarity scores are supplied to the matcher module. The matcher module combined these similarity scores and generated a descending list of freshers with their scores.

Algorithmically, first phase is written as follows:

```
Input: G, u
Output: E
1:  for each recently employed student g_i in G do
2:    Calculate similarity with the fresher
            s_u,g_i = S (u, g_i)
3:  end for
4:  Sort s_u,g_i in descending order
5:  Remove all entries below the threshold value
6:  Obtain top k recently employed students to form G_k
7:  Obtain employers to get employer set E related to G_k
8:  for each employer e_j in E do
9:    for each student g_i in G corresponding to e_j do
10:         if r_gi,e_j>0 then
```

```
11:                rate_acc= rate_acc + r_gi,e_j
12:          endif
13:          Rep(r_u,e_j)  = rate_acc/i
14:          sim_acc= w1 *S(u, gi)+ w2 * Rep (e_j)
15:    end for
16: end for
17: Sort E in descending order of sim_acc
18: return E
```

Algorithmically, second phase is written as follows:

```
Input: U, E
Output: U'
1: for each fresher u_i in U do
2:    for each recently employed student g_j of e in E do
3:          Calculate similarity S(u_i,g_j)
4:    end for
5:    M(u_i,e) = (M(u_i,e) + S(u_i,g_j))
6: end for
7: Sort M(u_i, e) in descending order
8: Obtain top N freshers to form U'
9: return U'
```

4 Results and Evaluations

To validate the proposed system, data from LinkedIn site is used. A dataset of 124 recently employed students at 20 companies having their profiles as courses (with a dictionary to map similar courses), 26 intern companies where a student have interned at, 23 intern titles i.e. the job role a student had been assigned during the internship, 11 institutions, 46 skills. These students' ratings about their companies (*https://www.glassdoor.com*) are also taken. Each company has a set of 15 assigned skills for which job openings are there and each of their employees can have minimum of 4 skills and a maximum of 8 skills. Moreover, Freshers' dataset of 105 unemployed students is taken from the campus itself.

A prototype was implemented for the proposed system and run on a dataset of 105 Freshers and 124 Recently Employed students with 20 companies. Recommendation list of 10 companies/employers for each fresher and recommendation list of 10 freshers for each employer is generated. To validate those lists, 10 test cases for first and second phase are executed and shown in the figures 2 and 3 respectively.

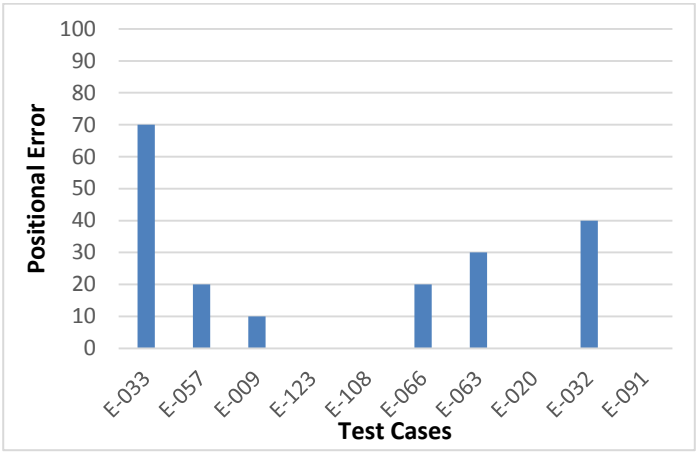

Fig 2: Percentage of error for employers' positions in the recommendation list for fresher

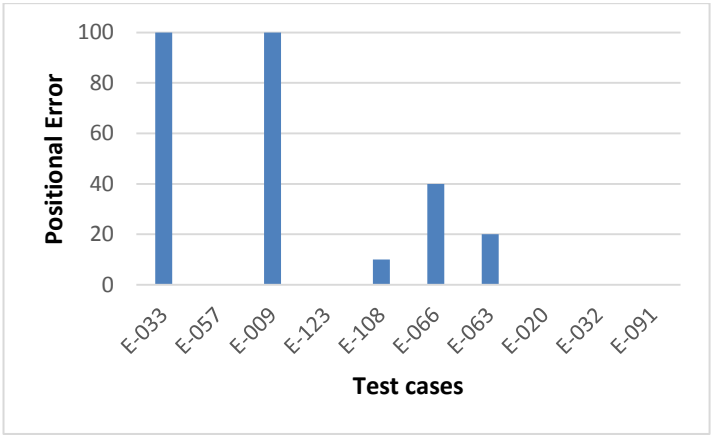

Fig 3: Percentage of error for freshers' positions in the recommendation list for company

Figure 2 and 3 represents the error present in the position of each of list items of recommendation lists generated for employer and fresher. It is found that, the system is **80%** accurate when recommending a company to a fresher and **75%** accurate when recommending a fresher to a company.

5 Conclusion

Understanding the need of fresher for an ideal job and employer for an ideal candidate, a job recommender system is proposed in this paper. The system used the similarity measures among fresher and recently employed students by the employers. These similarity scores are further refined using the reputation of the employers based on ratings by recently employed students. The system is implemented as a prototype and validated on the real data set as obtained from the LinkedIN site. The system performs better than similarity based recommender systems as it accounts for employer feedback through the rating mechanism.

6 References

1. Mamadou Diaby , Emmanuel Viennet and Tristan Launay :“Exploration of methodologies to improve job recommender systems on social networks“, Springer-Verlag Wien (2014).
2. Benjamin Kille and Fabian Abel :”Engaging the Crowd for better Job Recommendations ”, CrowdRec (2015)
3. Shaha T. Al-Otaibi and Mourad Ykhlef :“Job Recommendation Systems for Enhancing E-recruitment Process ”
4. Shengbo Shi : “Real-time Job Recommendation Engine Based on College Graduates' Personal ”,Journal of Residuals Science & Technology (2016)
5. Luiz pizzato, tomek rej, thomas chung, kalina yacefirena koprinska and judy kay : “reciprocal recommenders ”,university of sydney (2006).
6. Wenxing Hong, Siting Zheng, Huan Wang : “A Job Recommender System Based on User Clustering”, Journal of Computers (August 2013)
7. Rui Liu, Yuanxin Ouyang, Wenge Rong, Xin Song, Cui Tang, and Zhang Xiong : “Rating Prediction Based Job Recommendation Service for College Students”, Springer Journal (2016)
8. Rui Liu, Yuanxin Ouyang, Wenge Rong, Xin Song, Weizhu Xie, and Zhang Xiong: “Employer Oriented Recruitment Recommender Service for University Students”, Springer Journal (2016)
9. May Fern Koh, Yew Choong Chew: “Intelligent job matching with selflearning recommendation engine”, Elsevier Journals (2015)
10. Denis Gillet, Yao Lu, Sandy El Helou : “Analyzing User Patterns to Derive Design Guidelines for Job Seeking and Recruiting Website ” , École Polytechnique Fédérale de Lausanne (2012)
11. Shaha T. Al-Otaibi and Mourad Ykhlef : “A Survey of JobRecommender System” International Journal of the Physical Sciences(2012)
12. Denis Gillet, Yao Lu, Sandy El Helou : “A Recommender System for Job Seeking and Recruiting Website ” , École Polytechnique Fédérale de Lausanne (2013)
13. Deepali V Musale , Mamta K Nagpure, Kaumudini S Patil, Rukhsar F Sayyed : “Job Recommendation System Using Profile Matching And Web-Crawling ”,International Journal Of Advance Scientific Research And Engineering Trends (2016)