IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

Open Source Resume

Sheela Bankar[1], Ambesh Vishwakarma[2], Siddhi Sarolkar[3]

Assistant Professor, IT Dept, ISB&M SOT, PUNE, India [1]

Student, IT Dept, ISB&M SOT, Pune, India [2, 3]

Abstract: The Internet-based recruiting platforms become a primary recruitment channel in most companies. While such platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the opportunity of recruiting. The recommender system technology aims to help users in finding items that match their personnel interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. In order to improve the e-recruiting functionality, many recommender system approaches have been proposed. This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

Keywords: Open Source Software; Mining Software Repositories; Developer Expertise; Visualization Tool.

1. INTRODUCTION

Recently, job recommendation has attracted a lot of research attention and has played an important role on the online recruiting website. Different from traditional recommendation systems which recommend items to users, job recommender systems (JRSs) recommend one
type of users (e.g., job applicants) to another type of users (e.g., recruiters). In particular, job recommender system is designed to retrieve a list of job positions to a job applicant based on his/her preferences or to generate a list of job candidates to a recruiter based on the job requirements. To obtain a good recommendation results, many recommendation approaches are presented and applied in the JRS. Typically, given a user, existing JRSs employ a specific recommendation approach to generate a ranked list of jobs/candidates. However, different users may have different characteristics and a single recommendation approach may not be suitable for all users. Therefore, a high-quality JRS should have the capability of choosing the appropriate recommendation approaches according to the user's characteristic. We proposed an approach to construct and visualize developers' OSS biographies based on their contribution activities.

• Based on the approach, a visualization tool was implemented with another main feature of querying for OSS developers based on user's criteria.

• The tool is considered to provide a convenient way to seek suitable developers for constructing development teams in software development.

2. LITERATURE SURVEY

1. "A survey of job recommender systems,"
Summarized the categories of existing online recruiting platforms and listed the advantages and disadvantages of technical approaches in different jrss. For example, bidirectional recommendation is accomplished but only binary representation is allowed in the probabilistic hybrid approach. We also had done some research on feature extraction, resume mining, recommendation approach, ranking, and explanation for the JRS.

2. "Job recommender systems: a survey,"
User profiling and calculating similarity are presented as the prevailing process of a JRS, and the architecture and product features are briefly discussed. Moreover, empirical experiments had been conducted on a local online 1960 JOURNAL OF COMPUTERS, VOL. 8, NO. 8, AUGUST 2013 © 2013 ACADEMY PUBLISHER doi:10.4304/jcp.8.8.1960-1967 recruiting website and details on the specific case study are illustrated in Section IV.

3. "Resume information extraction with cascaded hybrid model," Presented a cascaded extraction approach for resumes to obtain the more effective information.

4. "Machine learned job recommendation," Trained a machine learning model to predict candidates' next job transition based on their past job histories as well as the data of both candidates and enterprises in the web.

3. RELATED WORK

Social Network Service(SNS) has been one of the most popular topics around the world since the last decade [1]. As a result of the rapid increase in SNS users, some studies applied data mining techniques to analyze SNS for exploring trending topics and seeking desirable community for oneself

• Content-based Recommendation
The principle of a content-based recommendation is to suggest items that have similar content information to the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

corresponding users. For example, in the recommendation that recommending jobs to a job applicant, the content is the personal information and their job desires. While recommending candidates to recruiters, the job description posted by recruiters, including the background description of enterprises, are used as the content for recommendation.

• Collaborative Filtering Recommendation
Collaborative filtering recommendation, known as the user-to-user correlation method, finds similar users who have the same taste with the target user and recommends items based on what the similar users like. The key step in CFR is computing the similarities among users. Collaborative filtering recommendation algorithm can be classified into memory-based and model-based. In the memory-based collaborative filtering recommendation, a user-item rating matrix is usually used as the input [2,3]. Applied in the job recruiting domain, some user behaviors or actions can generate the user-item rating matrix according to the predefined definitions and transition rules.

• Knowledge-based Recommendation
In the knowledge-based recommendation, rules and patterns obtained from the functional knowledge of how a specific item meets the requirement of a particular user, are used for recommending items [5]. For example, employees who have one or more years of work experience exhibit better performance as compared to those without experience.

• Reciprocal Recommendation
Firstly proposed by Luiz Pizzato et al. [4], reciprocal recommender is a special kind of recommender systems. The preferences of all the users are taken into account and need to be satisfied at the same time. A similarity calculation method for calculating the reciprocal value and achieving the reciprocal recommendation based on the explicit preferences obtained from users' resumes and the implicit preferences acquired from the user's interaction history.

4. PROPOSED SYSTEM

I. System introduction
We are developing system which is helpful for job seekers. Main advantage of this application is search is efficient. We have proposed new features which can classify category of user. User can select whether he/she is fresher or experienced. User can upload his/her resume which is shortlisted by admin according to company`s requirements. Admin will contact with H.R. executive. H. R. Will schedule interview for candidate and notify to admin. E-recruiting process is improved.

II. System module
1. Candidate profile
We have created login profile for candidate. Candidate will register to job portal by filling registration form. System will provide login ID and password to candidate. Candidate can sign in to job portal by entering given credentials. Candidatehave to fill all details while registering to job portal.

2. Admin login
This module is designed for admin. Admin will play important role in application. Admin will also have login credential to sign into account. Admin will store all information related to candidate in database. Admin will be intermediate between candidate and H.R. company. Admin will provide all details of candidate to H.R. Similarly, Admin will provide information of company to candidate. When candidate will apply to any job profile, admin will send candidate`s resume to H.R. Candidate will upload resume which is forwarded by admin to company. When candidate enter their keyskills, admin will search suitable company profile for candidate, and convey message to candidate. If candidate attend interview then admin will update his/her status.

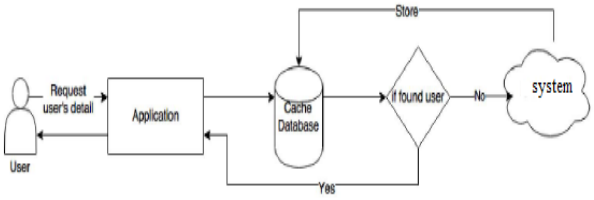
3. H.R. login
H.R module will be registered and possess credential as H.R. login. H.R. will possess information of candidate. H.R. will schedule interview for candidate. Details of further process will be maintained by H.R. There is another module i.e. company module. Candidate will search for various companies according to their requirements. All Company name and details will be stored in database.

4. Application
Candidate will apply through this module. Application of candidate will be stored. Number of companies and name, for which candidate is applying will be stored.

5. Interview portal
All scheduled and attained interview`s details will be shown in this module.

III. System features
- Efficient searching
- Reliable Review
- Proper resume building

Block diagram:-



Algorithm:
1. Levenshtein distance algorithm
2. levenshtein (source, target : STRING): INTEGER

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

```
3. -- Minimum number of operations to turn source into target
4.
5.      local
6.              distance : ARRAY_2 [INTEGER]
7.              i, j, del, ins, subst : INTEGER
8.      do
9.              create distance.make (source.count, target.count)
10.             from i := 0 until i > source.count loop
11.                     distance [i, 0] := i  ; i := i + 1
        end
12.             from j := 0 until j > target.count loop
                distance [0, j ] := j  ; j := j + 1
        end
13. from i := 1 until i > source.count  loop
14.             from j := 1 until j > target.count invariant
15.
16.             loop
17.                     if source [i ] = target [ j ] then
18.                             distance [i, j ] := distance [ i -1, j -1]
                        else
19.                             deletion := distance [i -1, j ]
                        insertion := distance [i , j - 1]
                        substitution := distance [i - 1, j - 1]
20.                     distance [i, j ] := minimum (deletion, insertion, substitution) + 1
21.                     end
22.                     j := j + 1
23.             end
24.             i := i + 1
25.     end
26.     Result := distance (source.count, target.count)
27. end
28.
```

K-means algorithm:-

- $x_1, \ldots, x_N$ are data points or vectors of observations
- Each observation (vector x_i) will be assigned to one and only one cluster
- C(i) denotes cluster number for the i^{th} observation
- Dissimilarity measure: Euclidean distance metric
- K-means minimizes within-cluster point scatter:

$$W(C)=\frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k}\left\|x_i-x_j\right\|^2=\sum_{k=1}^{K}N_k\sum_{C(i)=k}\left\|x_i-m_k\right\|^2$$

where
m_k is the mean vector of the k^{th} cluster
N_k is the number of observations in k^{th} cluster

- For a given cluster assignment C of the data points, compute the cluster means m_k:

$$m_k=\frac{\sum_{i:C(i)=k}x_i}{N_k},\ k=1,\ldots,K.$$

- For a current set of cluster means, assign each observation as:

$$C(i)=\arg\min_{1\le k\le K}\left\|x_i-m_k\right\|^2,\ i=1,\ldots,N$$

- Iterate above two steps until convergence

5. CONCLUSION

We presented state of the art of job recommendation as well as, a comparative study for its approaches that proposed by literatures. Additionally, we reviewed typical recommender system techniques and the recruiting process related issues. We conclude that the field of job recommendations is still unripe and require further improvements.

6. FUTURE SCOPE

As part of our ongoing research, we aim to build a new job recommendation approach and test with real data for employee and staffing data from large companies.

REFERENCES

[1] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," International Journal of the Physical Sciences, vol. 7(29), pp. 5127-5142, July, 2012.
[2] S. T. Zheng, W. X. Hong, N. Zhang and F. Yang, "Job recommender systems: a survey," In Proceedings of the 7th International Conference on Computer Science & Education (ICCSE 2012), pp. 920-924, Melbourne, Australia, July, 2012.
[3] K. Yu, G. Guan and M. Zhou, "Resume information extraction with cascaded hybrid model," In Proceedings of the 43rd Annual Meeting of the ACL, pp. 499-506, Ann Arbor, Michigan, June, 2005.
[4] Paparrizos, B. B. Cambazoglu and A. Gionis, "Machine learned job recommendation," In Proceedings of the fifth ACM Conference on Recommender Systems, pp. 325-328, Chicago, USA, October, 2011.
[5] M. Adedoyin-Olowe, M. M. Gaber, and F. T. Stahl. A survey of data mining techniques for social media analysis. CoRR, abs/1312.4617, 2013. II
[6] H. W. Ye, "A Personalized Collaborative Filtering Recommendation Using Association Rules Mining and Self-Organizing Map," Journal of Software, vol. 6(4), pp.732-739, 2011.
[7] L. Hu, W. B. Wang, F. Wang, X. L. Zhang and K. Zhao, "The Design and Implementation of Composite Collaborative Filtering Algorithm for Personalized Recommendation," Journal of Software, vol. 7(9), pp. 2040-2045, 2012.
[8] L. Pizzato, T. Rej, T. Chung, K. Yacef, I. Koprinska and J. Kay, "Reciprocal recommenders," In Proceedings of 8th Workshop on Intelligent Techniques for Web
[9] Personalization and Recommender Systems, held in conjunction with the 18th International Conference on User Modeling, Adaptation and Personalization (UMAP 2010), Hawaii, USA, June, 2010.
[10] R. Burke, "Hybrid recommender systems: survey and experiments," User Modeling and User-Adapted
[11] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," International Journal of the Physical Sciences, vol. 7(29), pp. 5127-5142, July, 2012.