

User Job Profile Matching Using Multiple Measures

¹Rashmi C, ²K S Adarsh, ³Hrithik V, ⁴Gururaj Suresh Babu, ⁵D. Yeshwanth Kumar Reddy

^{1,2,3,4,5}School of Computing & Information Technology, REVA University, Bangalore, India

¹rashmi.c@reva.edu.in, ²adarshks444@gmail.com, ³hrithikv20@gmail.com, ⁴hadligururaj@gmail.com, ⁵yeswanthreddy5033@gmail.com

Article Info

Volume 83

Page Number: 4921-4924

Publication Issue:

May - June 2020

Article History

Article Received: 19 November 2019

Revised: 27 January 2020

Accepted: 24 February 2020

Publication: 16 May 2020

Abstract

Social media has been increasingly used in various aspects of job matching with respect to user job profile, but all of it has been done manually by the Human Resource personnel. Multiple studies have already shown the effectiveness of social media analysis in job matching. Our goal here is to implement it in an effective and convenient system and thereby save time and resources wherever possible.

Keywords: job matching, social media analysis, modeling, weights

1. Introduction

Social media usage has been exponentially increasing along with the technological boom in Information Communication Technology (ICT). Social media data has various information useful to the recruiters which can be derived from their affiliations, projects, and friend circle. Currently these data have to be searched by the Human Resources Department manually and individually for each candidate, which takes more time and resources than an automated system.

This paper intends to add to the present works by deliberating a sample online user profile job matching and employment method that incorporates social networking data, as part of jobseeker modeling system for post matching and additional job endorsement.

The system will also help the job seekers who wouldn't be able to sort through the large amount of job vacancies and see if they qualify for the job, if they can work in the location specified, etc.

2. Working

A. Data Extraction

To start our matching process, we first need all the data about each job seeker and about the vacant jobs. This can be obtained in multiple ways. One way is to scrap data from social media directly using web scraping tools such as Robotic Process Automation or python. These tools scour the entire website searching for data and connection of the candidates apply. But this could prove extremely difficult to accomplish technically and may also violate privacy laws in most social media websites. To avoid such

issues, we recommend directly asking the users for this information and building our own social media database.

B. Data Processing

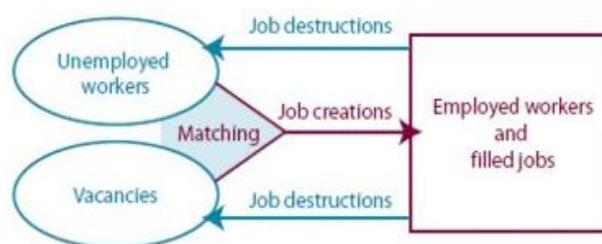


Figure 1: Job Flow

The data gathered from the above method will include the vacant job details such as; the location of the job, minimum qualifications, the job title, salary, etc.

These features shall be matched with the job seekers data to verify that the candidate qualifies for the job position i.e minimum educational qualification is met, the candidate is willing to work in the location describe in the job posting, etc.

After making sure all the conditions are met we get the probable job pool, which contains all the job seekers who technically qualify for the job. Now we shall rank these candidates using various techniques to ensure the best match is found.

C. Modeling

The job seeker's skills shall be measured using the three categorical system:

i. Subjective measure:

The subjective measure is taken by asking the user to rate themselves on the particular skill

ii. Semi – Subjective measure:

The semi – subjective measure is gathered from the social factors such as number of recommendations, project affiliations, etc.

iii. Objective measure:

The objective measure is the result of a short online quiz taken by the candidate or similar tests.

These measures when combined give an accurate score to the seeker’s skills.

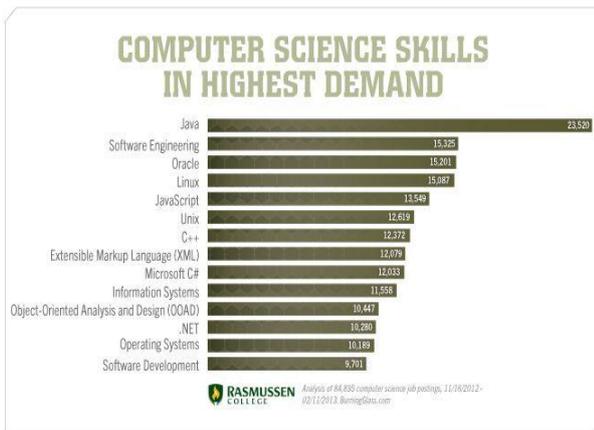


Figure 2: Analysis: Computer Science Job Posting And Skills

D. Weighting

The weights to the skills measures can be determined by the company as per their requirements. In our demonstration we assigned the highest weight to objective measure and the lowest weight to the subjective measure as we felt objective measures are the most important.

E. Ranking and Scoring

These weighted scores shall then be normalized to a score out of 100 which signifies how accurate of a match it is between the seeker and the job post. All the candidates who do not qualify will be assigned zero ‘0’ and the perfect match will have ‘100’.

However rarely will someone be a perfect match, so we should refer to the score to see who has got the highest score to determine the best fit.

3. Implementation

A. Turicreate

To implement the system, we use ‘turicreate’ which is an open source software tool for creating machine learning models and implementing them using various techniques, for jobs such as image classification, article discovery, flair transmissions, references, and more.

These software’s provide useful insights into the data



Figure 3: Job Posting Statistics

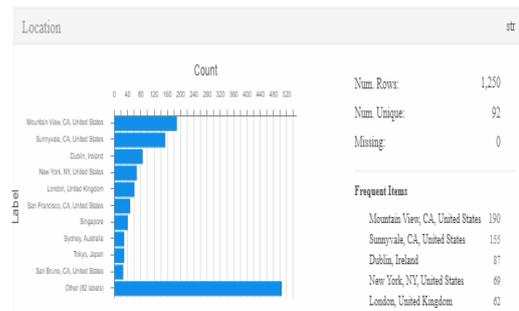


Figure 4: Job Location Statistics

We used Google’s job posting dataset to illustrate.

Turicreate uses SFrames as data objects so we shall have to first convert the dataset into SFrames. SFrame is an ascendable, core and frame based datagram, which accepts you to toil with datasets that are greater than the quantity of RAM on your computer. Using SFrames allows you to use enormous amount of data as you are no longer limited by the RAM of your system.

B. Python Notebook

(Previously identified as the IPython Notebook) The IPython Notebook is now recognized as the Jupyter Notebook. It is an cooperating computational system, in which you can conglomerate program implementation, rich script, calculation, designs and rich media. Python Notebooks will be used as the actual interface as they are highly portable and work on most systems efficiently.

```

Job Recommender System

Import turicreate
In [1]: import turicreate,

In [2]: # used to handle SFrames

Load Job Vacancy Dataset
In [4]: jobv = turicreate.SFrame('job_skills.csv')
Finished parsing file /home/ad/job_skills.csv
Parsing completed. Parsed 100 lines in 0.047933 secs.
.....
Inferred types from first 100 lines of file as
column_type_string[100:str,100:str,100:str,100:str]
If parsing fails due to incorrect types, you can correct
the inferred type list above and pass it to read_csv in
the column_type_hints argument
.....
Finished parsing file /home/ad/job_skills.csv
Parsing completed. Parsed 1250 lines in 0.046152 secs.

Load Job Seekers Dataset
In [5]: job = turicreate.SFrame('job_seekers_data.csv')
Finished parsing file /home/ad/job_seekers_data.csv
Parsing completed. Parsed 50 lines in 0.033274 secs.
.....
Inferred types from first 100 lines of file as
column_type_string[100:str,100:str,100:str,100:str,100:str,100:str,100:str,100:str,100:str,100:str]
If parsing fails due to incorrect types, you can correct

```

Figure 5: Snapshot of Vacancy and Job seekers dataset

C. Code

Functions used are:

- i. Function `setscore()` uses the data present in the jobseeker dataset to assess individuals and assign them a score out of 100 depending on the job position

```
def setscore(row,job_pos):
    if(row['Skill_1']==job_pos):
        if(row['Location']==job[jobv['Title']==job_pos]['Location']][0]):
            if(row['Qualification']==job[jobv['Title']==job_pos]['Minimum Qualifications']][0]):
                r= (row['Skill_1_x1']+(1.5*row['Skill_1_x2'])+(2*row['Skill_1_x3']))/4.5;
                return r
    if(row['Skill_2']==job_pos):
        if(row['Location']==job[jobv['Title']==job_pos]['Location']][0]):
            if(row['Qualification']==job[jobv['Title']==job_pos]['Minimum Qualifications']][0]):
                r= (row['Skill_2_x1']+(1.5*row['Skill_2_x2'])+(2*row['Skill_2_x3']))/4.5;
                return r
    if(row['Skill_3']==job_pos):
        if(row['Location']==job[jobv['Title']==job_pos]['Location']][0]):
            if(row['Qualification']==job[jobv['Title']==job_pos]['Minimum Qualifications']][0]):
                r= (row['Skill_3_x1']+(1.5*row['Skill_3_x2'])+(2*row['Skill_3_x3']))/4.5;
                return r
    else:
        return r
```

Figure 6: Snapshot of `setscore()` code

- ii. Function `getapplicants` sorts the applicants according to the scores assigned to each candidate for a particular job , and displays the result in a visual graph.

```
def getapplicants(job_position):
    jobs['score'] = jobs.apply(lambda x : setscore(x, job_position))
    jobs.sort('score',ascending =False)
    turicreate.show(jobs['Name'],jobs['score'])
```

Figure 7: Snapshot of `getapplicants()` code

D. Parameters Considered

User Job profiles are used to consider the multiple measures for job matching, in the methodology we make use of the following parameters as different measures for user profile based job matching.

- i. Number of skills and their expertise.

- A. Subjective Rating.
- B. Semi – Subjective Rating.
- C. Objective Rating

For each skill

- ii. Location where the candidate is willing to work.
- iii. Educational and vocational qualifications.

4. Result

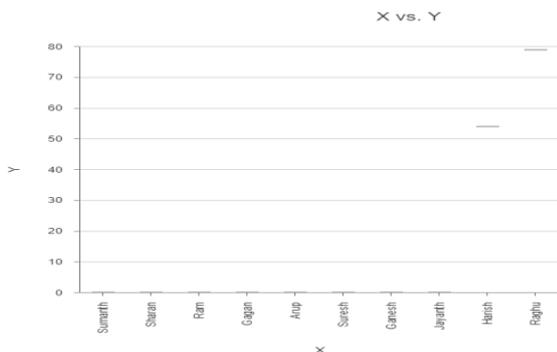


Figure 8: Final result indicating performance of each job seeker

The company will only have to specify the job title and the system will retrieve the data from the vacant jobs dataset and will give this visually intuitive graph as an output.

The users can also request the output in a tabular format.

Skill_3_x3	score	id
60	78.8888888889	3
None	53.8888888889	6
80	0.0	1
90	0.0	2
80	0.0	4
80	0.0	5
None	0.0	7
None	0.0	8
None	0.0	9
None	0.0	10

Figure 9: Score of job seekers sorted

Name	Location	Qualification	Skill_1	Skill_1_x1	Skill_1_x2
Sumanth	Singapore	BA/BS degree or equivalent practical ...	Google Cloud Program Manager ...	90	80
Sharan	Shanghai, China	BA/BS degree or equivalent practical ...	Supplier Development Engineer (SDE) ...	90	90
Raghu	Mountain View, CA, United States ...	Programming experience in one or more of the ...	Developer Advocate	75	80
Ram	Sunnyvale, CA, United States ...	BA/BS degree or equivalent practical ...	Audio Visual (AV) Deployments ...	85	75
Ganesh	Dublin, Ireland	Ability to speak and write in English and ...	Global Customer Experience ...	80	70
Harish	Mountain View, CA, United States ...	Programming experience in one or more of the ...	Supplier Development Engineer ...	90	95
Gagan	Sunnyvale, CA, United States ...	BA/BS degree or equivalent practical ...	Strategic Technology Partner Manager ...	80	85
Jayanth	Xinyi District, Taiwan	BA/BS degree in Engineering, Supply C ...	Manufacturing Business Manager ...	75	90
Suresh	New York, NY, United States ...	Experience in writing software in one or more ...	Solutions Architect	70	80
Arup	Xinyi District, Taiwan	BA/BS degree in Engineering, Supply C ...	Manufacturing Business Manager ...	80	75

Figure 10: Job Seeker Dataset

5. Conclusion

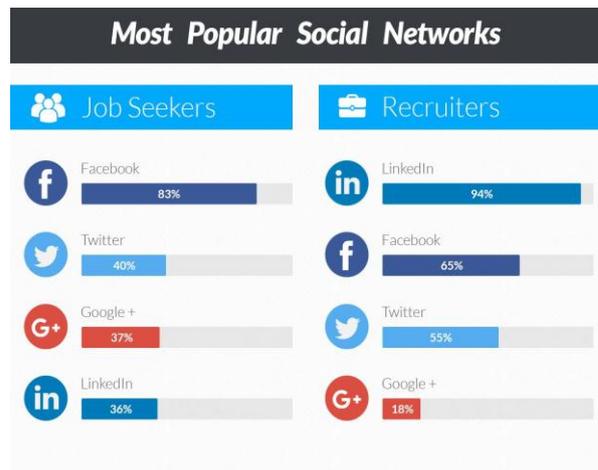


Figure 10: Important Social Networks for Job Matching

Our system will help both the job seekers and the companies posting jobs, by reducing the time spent on the user profile job matching phase and providing accurate results. The system could be used in a number of ways such as in internal mobility within a company (promotion - demotion), in external mobility (hiring - firing) and as a baseline for students to aim at to prepare exactly for their dream jobs.

References

- [1] Panniello, Umberto and Tuzhilin, Alexander and Gorgoglione, Michele. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and UserAdapted Interaction*, Volume 24, Number 1-2, Pages 35-65, Springer. 2014.
- [2] Chala, S.A, Ansari, F., and Fathi, M. (2016). Towards Implementing Context-aware Dynamic Text Field for Webbased Data Collection, *Int. Journal of Human Factors and Ergonomics* [in press]
- [3] Toteva, Kostadinka and Gourova, Elissaveta. (2011) Social Network Analysis in Professional e-Recruitment. *Third International Conference on Software, Services and Semantic Technologies S3T 2011*.
- [4] Pérez-Rosés, Hebert and Sebé, Francesc and Ribó, Josep Maria. Endorsement deduction and ranking in social networks. *Journal of Computer Communications*, Volume 73, Pages 200—210, Elsevier. 2016
- [5] Otte, Evelien, and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of information Science* 28.6 (2002): 441-453.
- [6] Morgan, Robert L. Job matching: Development and evaluation of a web-based instrument to assess degree of match among employment preferences. *Journal of Vocational Rehabilitation*, Volume 29, Number 1, Pages 29—38, IOS Press. 2008
- [7] Garg, Rajiv and Telang, Rahul. To Be or Not To Be Linked on LinkedIn: Job Search Using Online Social Networks. 2011
- [8] <https://www.smartjobboard.com/blog/job-board-social-media-marketing/>
- [9] <http://www.rasmussen.edu/degrees/technology/blog/computer-science-vs-information-technology-skills-you-need/>
- [10] <https://www.linkedin.com>
- [11] Jorge Valverde-Rebaza Ricardo Puma Paul Bustios Nathalia C. Silva “Job Recommendation based on Job Seeker Skills: An Empirical Study” 2018