Job Matching and Pushing Software System

Project Plan

Supervisor
Dr. Anthony Tam

Student
Tsoi Tsz Keung
# Table of Content

Abstract .................................................................................................................. Page 3  
Background ............................................................................................................ Page 3  
Problem Identification ............................................................................................. Page 4  
Objective and Scope ............................................................................................... Page 5  
Methodology ........................................................................................................... Page 11  
Risk and Challenges ............................................................................................... Page 13  
Schedules .............................................................................................................. Page 14
Abstract
The final year project is to build a job matching and pushing software system which aims at simplifying the traditional recruitment process and optimizing the existing job matching algorithm.

This paper will firstly review the existing recruitment software platform and identify the inefficiencies of them, secondly it will put forward an innovative matching algorithm and job pushing idea, then it will explain the methodology, analyze the risk and challenges, and list the tentative schedules of the project development.

Background
With the rapid development of Internet, online recruitment platform has become the mainstream tools for job seekers to hunt job and recruiters to find talents. Based on recruitment process, existing recruitment platforms could be approximately classified into three types:

1. Post and Apply
This is the most traditional one, recruiters post a position vacancy on the job board and job seekers can search and apply the job via sending email or directly contact with the recruiters. An example would be Interactive Employment Service of Labor Department (HKSARG).

2. Job matching
This type of recruitment software platform acts like a job agent. It usually requires users to register an account. For Job seekers, they may create his own profile and get matched with job recruitments while for recruiters, the system will find suitable candidates in its talents pool and recommend to them. An example would be Switch, Jobfox and RealMatch

3. Using Social Network
In this type of recruitment platform, it provides a social network platform for people to make his professional profile, make connection with each other and enlarge their social circle. It also maintains a job searching platform which allow users to search or post job using his professional profile. Besides, the social network platform helps users get attention from executive recruiters and companies can headhunt potential talents in the platform. LinkedIn Job would be an example.

Undoubtedly, these job recruitment platforms implemented their functionality and facilitated the job hunting and talents seeking process. However, they have deficiencies and the following session will identify them correspondingly.
Problem Identification

1. Time-consuming and Passive
The traditional job recruitment software platform (Post and Apply) is quite time-consuming and inefficient for both job seekers and recruiters.

For job seekers, they keep searching and browsing job posts, sending cover letter and resume for every single job application. These steps are time-consuming and dominate the whole recruitment process. For recruiters, they are in a passive role in finding talents, via posting job recruitment and waiting for people who are interested the job to apply. Under this recruitment method, recruiters have limited options and may not be able to find the one who fit their need the most.

2. Mismatching Problem
The existing job matching software system has limited success.

Among the current job matching algorithms, **Keyword search** and **Resume and Job Grading** are relatively popular and it is believed that they are not sufficient to determine whether a resume and job match or not. For **Keyword search**, the system usually parses the resume, omits unimportant tokens and common words, compare the resume with the job requirement to see how they match. This algorithm usually confronts with some intractable problems like misspelling, abbreviations, synonyms and hidden keyword issue, which result in having relatively high error rate. It is not enough, only using this method to determine the job matching. For **Resume and Job Grading**, the pre-defined weighting factor is complex and relatively subjective, which increase the uncertainty on how accurate the algorithm is.

3. Long-term Maintenance
The job recruitment platform, which make use of social network of user, usually need a relatively long-term maintenance before it can achieve its goal.

Take **LinkedIn Job** as example, user need to build a professional profile, make and maintain the connection with people. These would take considerable amount of time. Not until user successfully build a professional image can they obtain the attention from executive recruiters or apply job using their profile.
**Objective and Scope**

(1) Through data quantification and normalization, multidimensional modeling, crowdsourcing and machine learning optimization, the system aims at providing a more accurate job and resume matching algorithm.

(2) Adopting the job and resume broadcasting and pushing logic, the system allows job seeker’s resume to reach to all companies they may be interested in without applying and employers can get access to all potential talents who fit their need the most.

(3) Combined with well-built intermediary services, like Interview booking and tasks setting, the system helps simplifying and optimizing the job recruitment process.

### 1.1 Data Quantification and Normalization

There are different **common categories** of information in a resume and job recruitment, such as *educational background*, *working experience*, *skill set* and so on. For each resume and Job, the system can determine the relative position of them in a line, according to the content of the category it contains.

Take *education background* as an example, the system may take “primary school and above” and “Degree and above” as one end and middle position of the line. Take *working experience* as another example, the system may take “no related experience” and “five related experience” as one end and middle position of the line. However, for some category, the system need to take more steps to determine relative distance of resume and job. For example, *skill set*, the system might need to parse the content to each resume, using keyword search to calculate their similarity to the job recruitment, then using percentage to represent how a resume is matched to the job requirement. The system may use “0% match” and “100% match” as one end and another end of a line. Then the system will normalize each category, give them a proper max value and min value.

The system quantifies abstract resume and job components, scale and converts resumes and jobs into a point of a line representing a category of information.
1.2 Multidimensional Modeling

The system views each category of information as a **dimension** in a coordinate system.

When the system needs to perform job matching, it will have a list of simple shortlisted resumes and 1 particular job. The system will firstly construct a multidimensional model for the job, convert the job and the list of resumes into points in the coordinate system.

Then the system will calculate the relative distances between every resume and the job, and use it to rank the resumes. Using this ranking, the system will decide which resumes should be matched to a job recruitment. In principle, the shorter distance a resume is from the job, the higher chance the system will match it to a job recruitment.

This is **Phase one** of the system, the system will use this algorithm to perform job matching for a period of time. In this duration, the system will record down the result of the matching, store good matching and bad matching data for jobs. The system will use these data as training data for the **Phase Two** - machine learning optimization.

1.3 Crowd sourcing

As mentioned above, the system will crowdsourced training data from user input. Firstly, let’s explain how the system determines a matching is good or bad.

Assume that after the above algorithm, the system allocates 3 resumes to a recruiter. After consideration, the recruiter wants to offer an interview to 2 job seekers.
Among the 2 job seekers, only 1 job seeker is interested in the job. Then system would assume there are 1 good matching and 2 bad matching for this job and store them for this particular job.

Having these data, the system will convert them into training data for machine learning. The following explain how these will be converted and used for machine learning.

1.4 Machine Learning Optimization

Each job can be classified into different categories (Job category) according to the job type, requirement, offer, company scale and so on. The ranking is not important, the aim of this classification is to group "similar" jobs into one type.

In the system, each job category will be assigned a multidimensional coordinate system mentioned above.

The prefix letter of job category may be referred to industry and the digit may be job position type with different level.
Since it is assumed that multiple jobs could be grouped into one Job category, the system now can use the good matching and bad matching data sets collected to construct a training model for different job category.

Having the training model for every job category, the system is now ready to use the models to do future prediction of job matching.

When there is a new resume, and the system need to decide whether it would be a good match to a job or not. It will firstly find the corresponding training model of the job, project the resume on the coordinate system, use KNN (K Nearest Neighbor) algorithm to predict whether it would be a good match or not. Then it marks the result in the model, and matches the resume to the job if needed.
1.5 Advantages of the Algorithm

1. User engagement

One of the noticeable advantage of collecting and using user-input data is that the system will be more close to the current real-world recruitment condition, every decision user made is changing the system. Using user-input data ensures the system is up-to-dated.

2. More accurate

Our matching algorithm is based on one simple hypothesis: under the same category of job, if a portion of resumes are known to be good matching in that category, then the nearby (similar) resumes would be more likely to be a good matching rather than bad matching in the category. This theorem is generally true according to the probability.

Using this algorithm, the system classifies resumes and jobs instead of grading using pre-defined weighing factor designed by the programmer, it is believed less this kind of pre-defined setting, the higher accuracy of matching the system would make.

2.1 Job Pushing Idea

The following is the main flow of the recruitment process in job seekers and recruiters’ perspective, showing how the system broadcasts and pushes resumes and jobs.
2.2 Features

1. No need to search, no need to apply

Job seekers do not browse job posts in this web platform; they do not need to apply to any job. Reason is that searching job posts and sending CV and Cover letter are very time-consuming. Our software saves job seekers’ valuable time by broadcasting their resume to all companies which are recruiting the type of job the job seeker wants to do.

2. Higher chance of getting a better job

In the traditional job recruitment software, the number of companies a job seeker can apply is very limited. In our system, if the desirable type of job matches the position a company is hiring, the company will get the CV. It means all potential employers will receive the CV.

3. It is an offer of interview, not a suggestion

In the existing job matching system, they match job seeker to job recruitment, which is a suggestion made by its system. In our system, each job pushing is an offer of interview given by a company which has already read your CV, which is much more valuable than a suggestion.

4. Recruiters are no longer being passive and have high degree of control in talents selection

They have the initiative to select talents, choose the most desirable one(s) from a list of CV.

5. Only disclose the company information to people the recruiter is interested

In the traditional job recruitment platform, recruiters’ job post will be publicly displayed to everyone. While in our system, recruiters only disclose important company information and job detail to those potential employees.

3. Intermediary Services

1. Interview Booking

Once the job seeker accepts the offer of interview, the system allows him to input available time slots. The recruiter selects a time slot and the system will ask confirmation from the job seeker. Once the system has the confirmation from two parties, the system will send email to the two parties for exchanging the contact information.
2. Task/Question Setting
An optional intermediary function for recruiter, which allows them to setup a list of customized question or tasks for their job recruitment, requiring potential job seekers to finish the question or tasks before proceed to Interview booking.

**Methodology**

**System Architecture**

For the font-end, the system would contain a *Web Platform* and *mobile application* in Android and IOS platform while for the back-end, the system would need a *Server*, a *Database* and a *Notification Engine*. Noticed that the project would use the Google Firebase but not develop it.

**Process of Development**

Firstly, the project manager will conduct the *Vision and Scope document* and *Supplementary specification document* to identify the all the functional requirement and non-functional requirement of the system.

Secondly, the project manager will follow the order to develop the system:

Website ➔ Server ➔ Database ➔ Mobile Application ➔ Notification Engine

For the font-end project components, it will firstly complete the *Use case*, *Domain Model* and *Sequence System Diagram* to make sure that it is well-designed and organized. Then it moves on to *UI design*, using non-programming tools to draw the layout and navigation of the software, making a *Prototype*, which would be used to collect feedback from testers. After that it will jump to the *Coding part*, implement the UI and system functionalities.
While for Server and Notification Engine, it will also go through the Use case, Domain Model and Sequence System Diagram step, then it will jump to the Coding part, enable the communication with font-end system.

While for the Database, it will design the Relational Diagram and Table, SQL statements and construct the Database.

Finally, the project will step into Testing and Maintenance part to make sure all the required functionalities is completed.

**Coding Method**

In website development, programmer will use Html5 for website tag elements, CSS for Styling, JavaScript for manipulating the html elements, AJAX for back-end logic like communicating with the server, Bootstrap and JQuery for more advanced styling like responsive design. In terms of programming software, programmers will use Adobe Dreamweaver.

In Mobile application development, programmer will use Html5, CSS, JavaScript, Cordova. The usage of Html5, CSS and JavaScript is basically the same with website development. Cordova is for wrapping these three type of document into cross-platform mobile application. In terms of programming software, Adobe Dreamweaver and Cordova command line will be used.

In the Server and Notification Engine development, Java would be the main programming language and Eclipse would the programming software. In the Database development, MySQL would be the main programming software.

In the meanwhile, the programmer will keep using Source Tree to do versioning of the program. For UI design, object design, the programmer will use InVision and Draw IO.
Risk and Challenges

1. Normalization and Scaling Factor
When the system normalizes features/categories of resume and job, it need to constraint them into certain range, give each of them a proper max and min value. Also in scaling resume and job, there are different scaling method like *Standardization* / *Min-Max Scaling*. How the system determines the ranges and the scaling method would affect how accurate the system is.

2. Classification of Job Category
As mentioned above, in order to make use of machine learning concept to do future prediction on whether a new resume will be a good match or not, the system need to group similar jobs into one category, and combine multiple set of good matching bad matching data into one set and use it to construct a training model for one particular job category. If this classification of jobs is too board, irrelevant data sets might be grouped into one group and the deviation of training model would be very large. Later one, the system would use some irrelevant data to do the future prediction and wrongly match resume to a job; While this classification is too narrow and specific, the total number of job category will be very big, and the system will need to construct and store so many training model for each one of them. Worst still, the system will need to collect much more good known data sets to train the learning models.

3. Validation Problem
(1) Matching Algorithm Validation
Assume the system has enough sets of good known data, the system will use certain percentage (X) of data to train the learning models, then use (1-X) percentage of data for testing and tuning the learning model. Finally, the system will mix the data and randomly select data for testing to see the error rate of the algorithm.

(2) Job Pushing Idea Validation
Assume the system has data about: On average, (1) The amount of time a job seeker need to take to get an offer of interview from a recruiter, (2) In certain duration of time, the number of offer of interview a job seeker would get. (3) The amount of time a recruiter would need to take to successfully hire an employee. Also assume the system has a group of testers of significant number, in terms of job seekers and recruiters. Then system can test and see how it enhances and facilitates the recruitment process by comparing the test result with the known statistical.

However, it is not easy to collect the above required data or resource for validation.
## Schedules

### Development process

<table>
<thead>
<tr>
<th>Duration</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9/2016 – 30/9/2016</td>
<td>Project plan</td>
</tr>
<tr>
<td></td>
<td>Project Website</td>
</tr>
<tr>
<td>1/10/2016 – 30/10/2016</td>
<td>Project common property and documents</td>
</tr>
<tr>
<td></td>
<td>Website development</td>
</tr>
<tr>
<td>1/11/2016 – 30/11/2016</td>
<td>Server development</td>
</tr>
<tr>
<td></td>
<td>Matching algorithm logic</td>
</tr>
<tr>
<td>1/12/2016 – 30/12/2016</td>
<td>Database development</td>
</tr>
<tr>
<td>1/1/2017 – 30/1/2017</td>
<td>Mobile application</td>
</tr>
<tr>
<td>1/2/2017 – 30/2/2017</td>
<td>Notification Engine</td>
</tr>
</tbody>
</table>

### Official Deadline

<table>
<thead>
<tr>
<th>Date</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/10/2016</td>
<td>Deliverables of Phase 1 (Inception)</td>
</tr>
<tr>
<td></td>
<td>1. Detailed project plan</td>
</tr>
<tr>
<td></td>
<td>2. Project web page</td>
</tr>
<tr>
<td>9/1/2017 – 13/1/2017</td>
<td>First presentation</td>
</tr>
<tr>
<td>22/1/2017</td>
<td>Deliverables of Phase 2 (Elaboration)</td>
</tr>
<tr>
<td></td>
<td>1. Preliminary implementation</td>
</tr>
<tr>
<td></td>
<td>2. Detailed interim report</td>
</tr>
<tr>
<td>16/4/2017</td>
<td>Deliverables of Phase 3 (Construction)</td>
</tr>
<tr>
<td></td>
<td>1. Finalized tested implementation</td>
</tr>
<tr>
<td></td>
<td>2. Final report</td>
</tr>
<tr>
<td>18/4/2016 – 21/4/2017</td>
<td>Final presentation</td>
</tr>
<tr>
<td>2/5/2017</td>
<td>Project exhibition</td>
</tr>
</tbody>
</table>