Job Matching and Pushing Software System

Interim Report

Student
Tsoi Tsz Keung 2013526596

Supervisor
Dr. Anthony Tam

22 January 2017
Abstract

Online job recruitment platform is one of the most prominent channels for both job seekers and recruiters to hunt jobs and find suitable employees. Over the past few years, new recruitment software systems have been developed for the sake of bringing about new recruitment experience to users. However, current software systems which follow the traditional recruitment pattern, do not analyze the semantic meanings of different components in resumes and job recruitments, or use non-adaptive matching algorithm may not be able to efficiently facilitate the recruitment process. This paper introduces an innovative recruitment solution – JobMP, a job matching and pushing software system using semantic analyzers, crowdsourcing and machine learning optimization to provide a more accurate and adaptive resume and job matching algorithm. Adopting a new job pushing approach, JobMP allows resumes to reach to companies automatically and recruiters can actively select potential employees. Combined with well-built intermediary services, JobMP would simplify the traditional recruitment process and optimize the existing job matching algorithm. Over 300,000 resumes and 1,000,000 job posts will be used for the semantic analyzers training. The project is still under development, one of the main deliverables - web platform is completed and the overall matching algorithm is designed. The project will progress into the mobile application development and the machine learning optimization.

Keywords: job matching, semantic analysis, crowdsourcing, machine learning, intermediary services

Acknowledge

I would like to express my deepest gratitude to all those who have helped or motivated me in the project. A heartfelt appreciation I give to the project supervisor, Dr. Anthony Tam, who inspired and encouraged me during the project development, contributed ideas, pointed out the potential defects of the system and gave suggestions for improvement. He enriched the project substance and enhanced the project to the next level.

Furthermore, I would also like to appreciate Kaggle and CareerBuilder which provided resumes and job recruitments data for the model training and algorithm testing in this project.
List of Figures

Figure 1. Four sample resumes data in CSV file .................................................. Page 9
Figure 2. One job post sample in CSV file .......................................................... Page 9
Figure 3. PositionName2Vec Analyzer usage 1 .................................................. Page 9
Figure 4. PositionName2Vec Analyzer usage 2 .................................................. Page 9
Figure 5. SkillName2Vec Analyzer usage .......................................................... Page 9
Figure 6. Sample output of PositionName2Vec Analyzer .................................. Page 9
Figure 7. Data conversion of Education Level, Major, Industry, position name, duration .... Page 11
Figure 8. Data conversion of Skill set, Salary range, wok location, work time, company scale .... Page 11
Figure 9. relative position of resumes and job in Education Background Category ......... Page 11
Figure 10. Overall position of resumes and job ............................................... Page 11
Figure 11. Crowd sourcing example part 1 ...................................................... Page 12
Figure 12. Crowd sourcing example part 2 ...................................................... Page 12
Figure 13. Job category cluster analysis .......................................................... Page 12
Figure 14. Job category coordinate system ...................................................... Page 12
Figure 15. Assembly matching data in one job category .................................. Page 12
Figure 16. Project new resume ..................................................................... Page 13
Figure 17. Predict new resume matching result ............................................ Page 13
Figure 18. Job pushing workflow ................................................................. Page 13
Figure 19. System Architecture ................................................................. Page 13
Figure 20. Proposed schedules and current progress .................................. Page 13
Figure 21. Screen shots of the web platform ............................................... Page 13

List of Tables

Table 1. Necessary data for matching. .......................................................... Page 10
# Table of Content

- Background ........................................... Page 5
- Objectives and Scope ................................ Page 8
- Job Matching Algorithm .............................. Page 9
- Job Pushing Idea ....................................... Page 19
- Intermediary Services ................................. Page 21
- Methodology ............................................ Page 21
- Difficulties and Challenges .......................... Page 24
- Current Progress ...................................... Page 25
- Conclusion ............................................. Page 26
- Reference .............................................. Page 27
Introduction
Job recruitment is a process of selecting and hiring the most suitable candidate for a job opening, which is an activity linking those looking for jobs and those with jobs [1]. Job recruitment in a timely and cost-effective manner is a common goal for the above two parties. A practical recruitment solution can help job seekers easily obtain recruitment opportunities and reduce the time they need to spend, also free recruitment companies from resources allocated in advertisement and screening.

With the rapid development of Internet, online recruitment platform has become the mainstream channel for job seekers to hunt jobs and recruiters to find suitable employees. Over the past few years, numerous software system has been developed and claimed as a solution to the recruitment problem. However, some are of the view that the traditional recruitment platforms are inefficient [2] while some believed that the technologies and algorithms used in current recruitment systems are insufficient to achieve satisfactory performance in related to job searching and resume analysis [3].

This paper introduces an innovative recruitment solution – JobMP, a job matching and pushing software system which aims at simplifying the traditional recruitment process and optimizing the existing job matching algorithm. JobMP puts forward a more accurate and adaptive resume and job matching algorithm, allows resumes to reach to companies automatically and recruiters can actively select potential employees, and provides well-built intermediary services such as interview booking.

The paper will firstly review the existing recruitment software platforms and identify the deficiencies of them, secondly it will explain the proposed matching algorithm, job pushing idea and intermediary services, followed by the design, implementation and the system architecture of the software system. Finally, it will analyze the difficulties and challenges, indicate current progress and plan for future development.

Previous work
Based on recruitment process, the existing recruitment platforms could be approximately classified into three types:
1. Post and Apply
The first type is the most traditional recruitment system, recruiters post a position vacancy on the job board and job seekers can search and apply the job via sending email or direct contact with the recruiter. An example would be Interactive Employment Service of Labor Department, which is a web platform provided by the Hong Kong government, to facilitate the job recruitment process.

2. Job matching
The second type of recruitment software platform acts like a job agent. It usually requires users to register an account. For Job seekers, they may create their own profile and get matched with job recruitments while for recruiters, the system will recommend suitable candidates to them. An example would be Switch, Jobfox and RealMatch. Some are cross-platform while some are particular designed for mobile application.

3. Using Social Network
This type of recruitment platform provides a social network platform for people to make their 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.

Problem Identification
Undoubtedly, the above job recruitment platforms implemented their functionality and facilitated the job hunting and talents seeking process. However, they contain potential deficiencies and this session will explore them correspondingly.

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 need to search and apply for every single job they are interested in. These steps are time-consuming and dominate the whole recruitment process. For recruiters, they are in a relative passive role in finding candidates, 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 performance of existing job matching software system still is not satisfactory enough [3][4].

Most current job matching software system is a recommender system. The followings are related works: Yao Lu, Sandy El Helou and Denis Gillet[6] developed a hybrid job recommender system which models a directed weighted graph in which edges represent content-based and interaction-based relation and entities represent job seekers, employers and jobs. It ranks jobs to candidate with reference to their relevance to target candidate, considering the preferences of both employers and candidates. Content-based similarity computation and a PageRank-style ranking algorithm were used in the system. Wenxing Hong, Siting Zheng, Huan Wang and Jianchao Shi[7] developed a job recommender system which group job applicants into different clusters based on their individual information and historical behaviors, and apply different recommendation strategies to users of different clusters. Deepali V Musale, Mamta K Nagpure, Kaumudini S Patil and Rukhsar F Sayyed[8] developed a job recommendation system which applies semantic matching, tree-based knowledge matching and query matching techniques. It also uses the web crawling method to obtain recruitment data for its keyword search for job profiles.

Among the current job matching system, not much of them emphasizes on how to ensure their recommendation system will always be up-to-dated, and how to solve some intractable semantic problems like misspelling, abbreviations, synonyms and hidden keyword issue while performing its keyword searching.

3. Long-term Maintenance

The job recruitment platform, which makes use of social network of user, usually requires 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.
Objectives and Scope

1. A More Accurate and Adaptive Matching Algorithm

*JobMP* aims at providing a more accurate and adaptive job and resume matching algorithm which can keep pace with the changing recruitment tendency. By using well-trained semantic analyzers and parser to analyze different components of resumes and job recruitment, the system projects shortlisted resumes and the target job recruitment onto a multidimensional coordinate system, uses their relative distance to calculate the similarity between resumes and job, as a ranking which enable the system to recommend suitable resumes to the target job recruitment. Further optimized by crowdsourcing and machine learning concept, *JobMP* would provide an accurate and adaptive job and resume matching algorithm.

2. Job Pushing Idea

*Job pushing* describes the workflow of the overall recruitment process in *JobMP*. The idea is that the system will broadcast job seekers’ resume to all companies they may be interested in so that no application is needed. Employers can get a list of suitable candidates recommended by the system so they can actively select potential employees who fit their need the most. *Job pushing* also means job seekers received a request of complete version of resume, offer of tasks or offer of interview sent from recruiters while *CV pushing* means the system recommend candidates to recruiters, forward a complete version of resume, task result, or acceptance of offer of interview from job seekers.

3. Intermediary Services

Combined with well-built intermediary services, such as Interview booking and tasks setting, *JobMP* helps simplifying and optimizing the job recruitment process.
Job Matching Algorithm

This session describes the detailed matching algorithm mentioned in Objective One of the software system in term of data description, semantic analyzers training, feature identification and extraction, data analysis, to crowdsourcing, machine learning usage and Testing.

Data Description

Over 300,000 resumes and 1,000,000 job posts data, provided by Kaggle and CareerBuilder, will be used for semantic analyzers training. The following figures are sample of resumes and job posts.

<table>
<thead>
<tr>
<th>UserID</th>
<th>City</th>
<th>State</th>
<th>Country</th>
<th>DegreeType</th>
<th>Major</th>
<th>GraduationDate</th>
<th>WorkHistoryCount</th>
<th>TotalYearsExperience</th>
<th>CurrentlyEmployed</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>Las Vegas</td>
<td>NV</td>
<td>US</td>
<td>Master's</td>
<td>Anthropology</td>
<td>1/1/2011 0:00</td>
<td>10</td>
<td>8</td>
<td>Yes</td>
</tr>
<tr>
<td>80</td>
<td>Astoria</td>
<td>NY</td>
<td>US</td>
<td>Master's</td>
<td>Journalism</td>
<td>1/5/2007 0:00</td>
<td>3</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>123</td>
<td>Baton Rouge</td>
<td>LA</td>
<td>US</td>
<td>Bachelor's</td>
<td>Agricultural Business</td>
<td>1/5/2011 0:00</td>
<td>1</td>
<td>9</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 1. four sample resumes data in CSV file

Figure 2. One job post sample in CSV file

Semantic Analyzers and Parser

The system would makes use of the following semantic analyzers and parser to complete the data pre-processing step of matching.

1. PositionName2Vec Analyzer

Figure 3. PositionName2Vec Analyzer usage 1

Figure 4. PositionName2Vec Analyzer usage 2
**PositionName2Vec** determines whether two position names are in the same cluster or not, and finds out the semantic similarity between two position phrases. It passes two position names and output a Boolean value or the similarity in unit of percentage.

### 2. Skill Set Parser and SkillName2Vec Analyzer

![SkillName2Vec Analyzer usage](image)

**Skill Set Parser** passes two skill sets paragraphs from job post and resume, omits punctuation, extra space and stops words, convert to lower case, perform tokenizing and stemming. Afterwards it remains two sets of skill keyword. The system then use **SkillName2Vec** analyzer to compare every pair of keywords in the two keyword list and output the average similarity of these two skill sets. The method of training the above four semantic analyzers are completely data-driven, using the resumes and job post data mentioned above, Natural Language Processing (NLP) method – **word2Vec** and Cluster analysis method - **K-mean method**.

**Word2Vec**

Word2Vec is a text mining technique that convert word into vector. It takes reference with nearby words to extract the conceptual meaning of words. It is an unsupervised machine learning technique. Word2Vec mainly use the Cosine Similarity to calculate the similarity between two vectors.

\[
\text{similarity} = \cos(\theta) = \frac{A \cdot B}{||A|| ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}
\]

**K-Mean Cluster Analysis**

K-mean method constructs a partition of a set of objects of different features value into a set of k cluster. Each cluster contains words of similar feature values. The main algorithm is that each cluster is represented by the center of the cluster, the system partition words into k nonempty subsets, calculates the centroids of cluster, assign each word to the cluster with the nearest centroids, re-calculate and re-assign until no more new assignment.
Sample Output

<table>
<thead>
<tr>
<th>Word Phrase 1</th>
<th>Word Phrase 2</th>
<th>Similarity</th>
<th>IsSameCluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software developer</td>
<td>Software Engineer</td>
<td>0.90128</td>
<td>true</td>
</tr>
<tr>
<td>Software developer</td>
<td>Java Programmer</td>
<td>0.710605</td>
<td>true</td>
</tr>
<tr>
<td>Software developer</td>
<td>Accountant</td>
<td>0.10724</td>
<td>false</td>
</tr>
<tr>
<td>Software developer</td>
<td>Clerk</td>
<td>0.07065</td>
<td>false</td>
</tr>
</tbody>
</table>

Figure 6. Sample output of PositionName2Vec Analyzer

Feature Identification

There are at least four common Categories of information between resumes and job recruitments. The following table shows the necessary data the system need to collect to perform the matching.

<table>
<thead>
<tr>
<th>Job seekers</th>
<th>Recruiters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resume - Educational Background</strong></td>
<td><strong>Resume requirement - Education</strong></td>
</tr>
<tr>
<td>1. Level of education:</td>
<td>1. Min level of education</td>
</tr>
<tr>
<td>2. Sub-info. under the Level</td>
<td>2. Sub-info. under the Level</td>
</tr>
<tr>
<td><strong>Resume - Working Experience</strong></td>
<td><strong>Resume requirement - Working Experience</strong></td>
</tr>
<tr>
<td>1. List of working experience</td>
<td>1. Related working experience:</td>
</tr>
<tr>
<td>1a. Industry</td>
<td>1a. Industry</td>
</tr>
<tr>
<td>1b. Position name</td>
<td>1b. Position name</td>
</tr>
<tr>
<td>1c. Number of year</td>
<td>1c. Number of year</td>
</tr>
<tr>
<td><strong>Resume - Skill set</strong></td>
<td><strong>Resume requirement - Skill / Preference</strong></td>
</tr>
<tr>
<td>1. List of skill set</td>
<td>1. List of skill requirement</td>
</tr>
<tr>
<td>1a. one skill</td>
<td>1a. one skill</td>
</tr>
<tr>
<td><strong>Requirement to job</strong></td>
<td><strong>Job nature</strong></td>
</tr>
<tr>
<td>1. Salary range</td>
<td>1. Salary range</td>
</tr>
<tr>
<td>2. Work location</td>
<td>2. Work location</td>
</tr>
<tr>
<td>3. Work time</td>
<td>3. Work time</td>
</tr>
<tr>
<td>4. Company scale</td>
<td>4. Company scale</td>
</tr>
</tbody>
</table>

Table 1. Necessary data for matching
The data collected from the two parties is symmetric in a sense that each category of information from the job seekers/recruiters can correspond to another one from recruiters/job seekers. Having this characteristic of problem, the system would like to find out how similar or different resumes and a job are.

**Feature Extraction**

In each resumes and job matching, there would be a list of resume and one particular job (target job). The purpose of this data pre-processing step is to filter those obviously irrelevant resumes thereby save system resources from performing unnecessary matching algorithm on these resumes. Another objective of these step is to convert the data collected previously, which is of different type, into numerical type so that the next matching algorithm can be conducted.

**Step 1: Filtering**

Each resume is associated with one or more than one desirable type of job, namely *Position Name*. The system uses the semantic analyzer, *PositionName2Vec* to find out whether the position name(s) of a resume and the position name of the target job are in the same cluster or not. If not, the system immediately filters out those resumes and will not perform matching algorithm on them later.

**Step 2: Extract Numerical Data**

The following figure demonstrates how the system converts data of different types into numerical data.

---

*Figure 7. Data conversion of Education Level, Major, Industry, position name, duration*
Figure 8. Data conversion of Skill set, Salary range, work location, work time, company scale

**Matching Algorithm**

After the above data pre-processing step, the system calculates the relative distance between each resume and the target job.

By viewing each common category of resumes and job recruitments as a *Dimension* in a coordinates system, the system forms a *Multidimensional Coordinate System* in which each axis represents a category of information and each space can represent a resume or job recruitment.

For each category of information, the system uses *Max-Min Normalization* method to adjust the scaling of data, and projects resumes and the target job onto the multidimensional coordinates system. Then the system calculates the *Euclidean Distance* between each resumes and the target job.
The system uses these relative distances to rank the resumes. Using this ranking, the system would decide which resume 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 the job.

In conclusion, the system quantifies abstract resume and job recruitment components, scales and projects them onto a multidimensional coordinates system and determines the relative position of them.

However, the above algorithm has some deficiencies. Firstly, it may not be realistic. It is possible that for some categories of job, some features, for example working experience, are more important than other features. Secondly, this algorithm is rule-based, fixed. Nonetheless, the recruitment condition and tendency are changing every day.

In views of the abovementioned problems, the system need a more adaptive matching algorithm, an algorithm which can learn from the past matching results, feedback from users, enable the system to keep pace with the changing recruitment tendency.

Here is the idea: The system would use the above matching algorithm, which is already working, to collect real-world resume and job matching data as training data, use them to develop another matching algorithm with machine learning optimization

**Crowd sourcing**

As mentioned above, the system will crowdsourse training data from user input. The followings show 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 optimization.

**Machine Learning Optimization**

**Job Classification**

Using *Cluster Analysis technique - Linear Discriminant Analysis* (LDA), each job can be classified into different categories (*Job category*) according to the job type, requirement, offer, company scale and so on. 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 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.

**Principal Component Analysis (PCA)**

The current system assumes the weights of the common categories of resumes and job recruitment are of the same. However, in reality, some types of job may value certain category over another one. Since the system has already collected sufficient good matching and bad matching data, it could perform a Principal Components Analysis (PCA) for each job category to find out the best weights for each category of information to optimize the prediction result.

**Future Prediction**

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 category the job belongs to, then project the resume on the coordinate system, use K Nearest Neighbor (KNN) 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.
Testing
At this stage, there is still no matching data available for testing. However, the following is the procedure for conducting test on the proposed algorithm.

Data
Assume that there is a list of job recruitment, Job₁...Jobₘ such that for each Jobᵢ, there is a list of resume of number GRᵢ which are labelled as good matching to that job, and a list of resumes BRᵢ which are labelled as bad matching to that job.

Notations
We will perform the job cluster analysis to classify m jobs into n job categories, n <= m. For each job category jobCᵢ, there exist good matching resumes GRCᵢ and bad matching resumes BRCᵢ, 1 <= j <= n. For example, if a job category jobC₁₀ contain job₃ and job₅, then GRC₁₀ = GR₃ + GR₅ and BRC₁₀ = BR₃ + BR₅. Also let the error rate of determining a resume in GRCᵢ as good matching be ErrGRCᵢ and let the error rate of determining a resume in BRCᵢ as bad matching be ErrBRCᵢ

Error rate
Firstly, the system calculates the Error rate for each job category.
Error rate $Errᵢ = (GRCᵢ \times \text{ErrGRCᵢ} + BRCᵢ \times \text{ErrBRCᵢ}) / (GRCᵢ + BRCᵢ)$
The total error rate of the algorithm $= \sum_{i=1 \text{ to } n} (GRCᵢ + BRCᵢ) \times \text{Errᵢ} / \text{(Total number of resumes)}$
This error rate would indicate whether the algorithm can determine resumes as correct type or not.
Advantages

1. Keep Pace with Changing Recruitment Tendency
One of the noticeable advantages of collecting and using user-input data is that the system will be more close to the current real-world recruitment condition, the decision user made will changing the system. Using user-input data ensures the system is up-to-dated.

2. More Accurate
Using semantic analyzers could increase the accuracy of the matching algorithm. The matching algorithm trained fours semantic analyzers to detect the similarity between phrases. It solves the misspelling, abbreviations, synonyms and hidden keyword issue. Besides, 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.
Job Pushing Idea

This session describes the job pushing idea mentioned in Objective Two of the software system in term of overall recruitment process from the perspective of job seekers and recruiters. The following graph demonstrates workflow of job pushing and CV pushing.

Figure 18. Job pushing workflow
Features

1. Time-saving
Job seekers do not search or browse job posts; They do not need to apply to any jobs. After completing a short version of resume and desirable type of job, all job seekers need to do is to wait for job pushing from system. Our software system 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. Therefore, the total amount of time job seekers spend in recruitment is reduced.

2. Higher Chance to Get a Better Job
In traditional job recruitment software systems, 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. Protect the Privacy of Job seekers
Instead of directly sending the complete version of resume to companies, the system will only send the short version of resume. If a recruiter wants to have more information of the job seeker, he must ask the job seeker for permission. This design of workflow protects job seekers from leaking private information to strange parties.

4. Recruiters have high degree of control in talents selection
They have the initiative to select talents, choose the most suitable one from a list of resumes.

5. Protect Recruitment Information
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 recruitment detail to those potential employees.
Intermediary Services

1. Interview Booking
Once the job seeker accepts the offer of interview, the system allows him to input available time slots for interview. 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 contains a Web Platform and Mobile Application in Android and IOS platform while for the back-end, the system contains a Server, a Database and a Notification Engine.
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 web platform and mobile application, 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 part to make sure all the required functionalities are completed. Testing includes the user interface under different dimension of devices, the communication between font-end system and back-end system, the protocols used in both sides are consistent or not.

Coding Method
In website platform development, project developers 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. HTML, CSS, JavaScript and AJAX are standard scripting language to develop web application. Bootstrap is chosen instead of Foundation because the number of different version of browser Bootstrap supports is larger than Foundation. The number of built-in animation library of JQuery is much larger than other JavaScript frameworks such as Node.js or AngularJs. In terms of programming software, project
developers will use *Adobe Dreamweaver* because it includes more useful embedded library than other programming software like *WebStorm* or *Microsoft WebMatrix*, it has a well-built debugging tools which allow the programmer fix bugs easily.

In mobile application development, instead of coding using *Android* and *Objective C*, project developers will use *Html5, CSS, JavaScript, Cordova*. The reason is that project developers need to develop the same mobile application twice using two scripting language in two different platforms if the *Android* and *Objective C* path is chosen. If project developers use *Html5, CSS, JavaScript, Cordova*, they can easily wrap the existing web platform into mobile application without further coding since *Cordova* enables them to convert web document into cross-platform mobile application. In developing cross-platform mobile application, *Cordova* and *PhoneGap* are available, the former one is chosen because *Cordova* is actually built based on *PhoneGap*, which means the project developers do not need to deal with low-level programming. 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. There are two ways to host a server, project developers can either setup the server in the computer owned by them, or rent a virtual machine owned by other commercial companies which provided cloud computing service. The latter option is chosen because project developers do not need to have a 24 hours’ maintenance on the computer. Also setting up a virtual machine in other company is less complicated because they have already taken care of those network security issue like *SSL*. Project developers will rent a virtual machine from *Google Cloud Platform* instead of *Microsoft Azure* because the price of renting one in Google is lower than in Microsoft and Google offers free trial version for three months. 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*. 
Difficulties 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 matching 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.
**Current Progress**

**Schedules**

<table>
<thead>
<tr>
<th>Proposed Schedule</th>
<th>Actual Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 2016</td>
<td>Sep 2016</td>
</tr>
<tr>
<td></td>
<td>- Project Plan</td>
</tr>
<tr>
<td></td>
<td>- Project Website</td>
</tr>
<tr>
<td></td>
<td>- Web Platform</td>
</tr>
<tr>
<td></td>
<td>- Matching Algorithm</td>
</tr>
<tr>
<td>Nov 2016</td>
<td>Nov 2016</td>
</tr>
<tr>
<td></td>
<td>- Server</td>
</tr>
<tr>
<td></td>
<td>- Machine Learning Optimization</td>
</tr>
<tr>
<td>Dec 2016</td>
<td>Dec 2016</td>
</tr>
<tr>
<td></td>
<td>- Database</td>
</tr>
<tr>
<td>Jan 2017</td>
<td>Jan 2017</td>
</tr>
<tr>
<td></td>
<td>- Mobile Application</td>
</tr>
<tr>
<td>Feb 2017</td>
<td>Feb 2017</td>
</tr>
<tr>
<td></td>
<td>- Notification Engine</td>
</tr>
<tr>
<td>Mar 2017</td>
<td>Mar 2017</td>
</tr>
<tr>
<td></td>
<td>- Machine Learning Optimization</td>
</tr>
</tbody>
</table>

Figure 20. Proposed schedules and current progress
Current Progress

At this stage, the Project Plan, Project Website, Web Platform, Overall Matching algorithm, Server and Database are completed.

![Figure 21. Screen shots of the web platform](image)

The development period of Web Platform and Overall Matching algorithm is longer than they were expected. However the Database development requires less time than it is planned.

The Machine Learning Optimization and Mobile Application are currently under development.

Conclusion

In this paper, it looked into the current job recruitment platforms and indicated the potential deficiencies of them, pointed out the three main functionalities of the JobMP, which are (1) more accurate job and resume matching algorithm (2) job and resume broadcasting and pushing concept (3) intermediary service like interview booking and task setting. The paper also described the system architecture of the software system and the methodology on implementation, analyzed the difficulties and challenges the project manager might encounter during the development. In the future development, the project manager will try to find solution and alleviate the risks as easily as possible and conduct more research on coding language and algorithm to ensure the good quality of the final product.
## Reference

<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
<th>URL</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[7]</td>
<td>Wenxing H., Siting Z., Huan W., and Jianchao S. A Job Recommender System Based on User. Available at:</td>
<td><a href="https://pdfs.semanticscholar.org/e2e4/a8e6887ea0e50ebeef372f72093a96cd25e0d.pdf">https://pdfs.semanticscholar.org/e2e4/a8e6887ea0e50ebeef372f72093a96cd25e0d.pdf</a></td>
<td>2016</td>
</tr>
</tbody>
</table>