

# Cluster based Ranking Index for Enhancing Recruitment Process using Text Mining and Machine Learning

Mayuri Verma  
Dayalbagh Educational  
Institute  
Agra-282005, India

## ABSTRACT

This paper presents an effective approach for extracting relevant words from the resumes using Term Document Matrix. The role of the candidate, various skills, familiarity with various frameworks, experienced skills and operating systems have been considered. A clustering methodology has been used to find the similar resumes. The importance of each word has been calculated according to the cluster which makes this paper unique. The appropriate rank of the resumes have been calculated. The experimental results shows that Cluster Based Ranking gives the potentially best candidate for a particular job profile. The weighted importance in calculating the ranks is the very first effort in itself. Further work can be done in this area for improving the productivity in the recruitment process.

## General Terms

Text Mining, Data Mining, Recruitment, Algorithms

## Keywords

Resume, K Means, ReliefF

## 1. INTRODUCTION

Technology plays a crucial role in any organization. When it comes to IT sector it becomes even more important because the aim of IT Sector is automation. With the large, rapidly growing IT sector, the recruiters receive a large number of applications daily. Selecting the right candidate for a job profile, with the help of an automated process is also very necessary in order to save time and manpower. There are many online job portals through which a candidate can apply for a particular role. Portals like LinkedIn [1], Monster.com [2], Naukari.com [3], Indeed.com [4] are very popular. It will be very time consuming if the recruiters go through each and every resume manually and then try to find the suitable candidates for the interview. Several studies have taken place in order to enhance the recruitment process. K Yu et al. presented a resume information extraction methodology with cascaded hybrid model [5]. The model depends on the structure of the resume and the information extraction is time consuming since for every section in the resume the algorithm uses different information extraction model which is dependent on the structure of that part of the resume. Chandola et al. developed an online resume parsing system using text analytics [6]. The system depends on a rating scale for individual keywords where according to the extracted words, low (1 point), medium (2 points) or high (3 points) are rated and then all these ratings are summed up to get the rank for that resume. In this work all the extracted words are assumed to be of equal importance but this is not the case. For different positions different skills have different priorities which this system doesn't take into consideration. Koppurapu et al. presented a method for automatic extraction of useful

information from unstructured resume and claimed high precision and recall [7]. A method for ranking the resumes has not been given by them. Several other efforts have been made by Zhi Xiang et al.[8], Zhang Chuang et al. [9], Celik Duygu et al. [10] for extracting the information from the resumes. Very little efforts have been made in the past in order to rank the resumes to find the suitable candidate.

Text Analytics is a rapidly growing field which can be defined as a statistical machine learning methodology for converting the unstructured raw information into structured information which can further be classified, mined, trained for better and high quality information. Text Analytics is also referred as Text Mining [11]. Methods of Pattern Recognition, Information Extraction, Data Mining, Parsing are involved in Text Mining and referred to as Natural Language Processing (NLP) [12]. In this work information extraction methodology has been developed and then Clustering algorithm has been used to find the similar resumes bases on the extracted words then a ranking methodology has been developed to find the suitable candidate.

The rest of the paper has been organized in the following sections - Section 2 describes about feature extraction from the resumes. In section 3 resume clustering has been described. In section 4 a cluster based ranking methodology has been described for ranking resumes. In section 5 results have been described and in section 6 conclusions and future work have been described. References have been given in section 7. The figure below shows the model proposed in this paper.

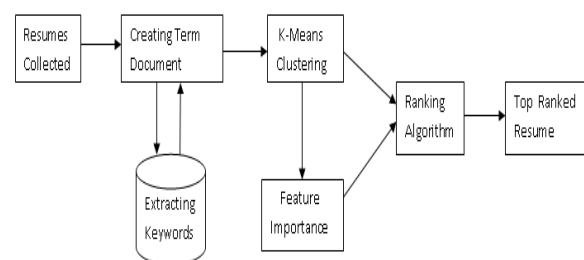


Figure 1: Proposed Model

## 2. FEATURE EXTRACTION FROM RESUMES

All the resumes have been obtained from Indeed.com [4]. In this work from the 574 resumes those words have been extracted which comes at least 75 times if addition of all the occurrences of that word is done in all the resumes. If there are 'n' resumes then for every distinct word 'w' in these

resume dataset , if the frequency count of ‘ w’ in a resume ‘ i’ is w(i) then:

$$\text{Frequency Count of word } W \text{ in the resume dataset (OFC)} = \sum_{i=1}^n W(i)$$

For those words W for which OFC is greater than 75 only those are considered in this study. In order to extract the words from the resumes and finding the OFC of a word across all resumes , resume Corpus has been made. In text mining a text Corpus is a large and structured way of storing large texts such that statistical analysis, pattern finding , rule mining , word occurrences and rule validation tasks could be performed easily. In this work the unstructured resumes have been created into Corpus with the help of "tm" package in R [13] by providing the Uniform Resource Identifier [14], which stores the information about the Corpus that all the resumes are provided to it as vector of all the resume file names . A method to read the text from the resume has been provided to the Corpus and thus a text Corpus of the resumes has been created. In order to find the OFC, term document matrix has been created. The term document matrix describes the frequency count of words among all resumes. In the term document matrix , each row represents one resume and each column represents a word and each entry represents the frequency count of a particular word in that particular resume. Some pre processing steps have been used to create the term document matrix. All the punctuation marks have been removed, all the stop words (A, An, The, Of, In , On etc al. ) have been removed since these are of no significant use , all the Upper case letters have been converted into lower case in order to maintain the similarity , all the numerical numbers have been removed and words are stemmed so that any differences in the tenses or the usage of the words would not be able to affect the term document matrix.

Various OFC values have been taken (50,75,100) and the results are compared before proceeding further with OFC=75. The comparison of results using various OFC values have been given in the table below.

**Table 1: Comparison of results using various OFC values**

Attribute	OFC=50	OFC=75	OFC=100
Number of Words in the term document matrix	606	427	345
Number of Relevant Words for Resume Ranking	83	63	54
Comparison	Words are biased towards specific resumes, skills are not generic and OFC is very low	Words are not biased towards specific resumes, Skills are generic	Very few words, Not been able to capture all generic skills

OFC value has been taken as 75. The relevant words are categorized on the basis of their impact on the resume. Role of the applicant is necessary in order to assess his background and his past work experience. Second category is taken because for any applicant in the application development industry, programming skills , Database management skills and Web related skills are must. In the third category the software packages, frameworks, platforms are there on which

the candidates have worked. There are some other skills which the candidate had developed other than the general skills through experience are taken in the next category. In the next category the operating systems are taken on which the candidate has worked. The relevant words along with the category for OFC=75 are given in the table below.

**Table 2: Category wise relevant words**

Category	Words
Role	Administrator, Analysis, Analyst, Application, Architecture, Backend, Client, Database, Debug, Descript, Design, Develop, Front, Intern, Network, Server, Web
Language, Database and Web related skills	CSS, HTML, Javascript, JQuery, Json, mySQL, php, plsql, Servlet, Shell, SQL, sqlite, XML
Software Packages, Tools and Frameworks	ADO.NET, AJAX, ASP, ASP.NET, Bootstrap, Eclipse, JSP, MVC, SDK, Visual, Wordpress, XCode
Experience Skill mentioned in the resume	API, App, Automation, Bug, Cloud, Code, Configuration, Core, Data, Feature, Module, Queri, Scripting, Software, Website
OS	Android, iPhone, Linux, Unix, Windows

### 3. CLUSTERING OF RESUMES

K-Means Clustering [15] has been used in order to cluster the resumes having the relevant words counts as features. K-Means Clustering algorithm is an Unsupervised learning algorithm in the field of Data Mining. Unsupervised learning is that type of learning in which the learning algorithm draws inferences from the provided dataset in which the data is not labeled. In this work all the resumes have been categorized into one of the K clusters and the resume goes into the cluster with the nearest mean. If there are ‘n’ resumes  $R^{(1)}, R^{(2)}, R^{(3)}, \dots, R^{(n)}$  so in order to cluster them we ,the training data as the relevant words count in each resume has been taken. Thus for every resume  $R^{(i)}$  , a feature vector of size 63 has been provided. Thus for every resume  $R^{(i)}$ , there is a feature vector  $F^{(i)} \in R^n$ , where  $i=1,2,\dots,n$  (number of resumes). The target of the K-Means Clustering Algorithm is to determine K centroids (one for each cluster) and assign a cluster  $K(i)$  to each resume  $R(i)$ . The algorithm works as follows:

1. Initialize Cluster Centroids  $M_1, M_2, M_3, \dots, M_k \in R^n$  randomly.

2. Repeat until it converges

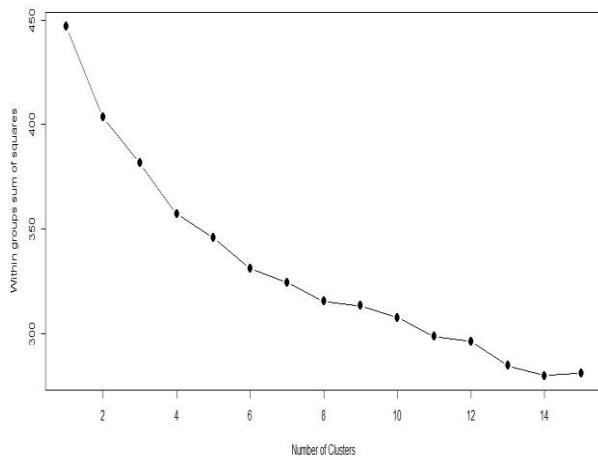
{  
For every resume  $i$ ,  $K(i) = \min_K ||R^{(i)} - M^{(K)}||^2$  ( Calculating the Euclidean Distance)

For each cluster  $K$ ,  $M_K = \frac{\sum_{i=1}^m \{ K^{(i)}=K \}. R^{(i)}}{\sum_{i=1}^m \{ K^{(i)}=K \}}$

}

The feature vectors are normalized so that the value of word count ranges only between [0,1]. In order to determine the number of clusters K, Elbow Method [16] has been used. In Elbow method the percentage of variance is looked as a function of the number of clusters K. Number of clusters K

have been chosen such that adding another cluster doesn't give much better modeling of the data. In the figure below the number of clusters for resumes has been determined using Elbow Method.



**Figure 2: Assessing the Optimal Number of Clusters with the Elbow Method**

From the Elbow Method, the conclusion was drawn that after K=10, adding another cluster doesn't add much significance. Thus the resumes have been divided into 10 clusters using K-Means Clustering Algorithm.

In order to find the important features in each cluster ReliefF algorithm [17] has been used. While finding important words for each cluster 'i', all cluster instances of cluster 'i' are labeled as 1 and all remaining are considered to be label 2 and then ReliefF algorithm has been used with these labels as the target values and the 63 words training vector as the feature. ReliefF works on dataset with n resumes, each having F features and the feature values must be normalized between 0 and 1. The algorithm repeats m times. The algorithm starts with F- long weight vector each initialized with 0. At each iteration, take the feature vector (X) belonging to one random resume from the dataset, and the feature vectors of the instance closest to X (by Euclidean distance [18]) from each class. The closest same-class instance is called 'near-hit', and the closest different-class instance is called 'near-miss'. The weight vector is kept on being updated such that:

$$W_i = W_i - (X_i - nearHit_i)^2 + (X_i - nearMiss_i)^2$$

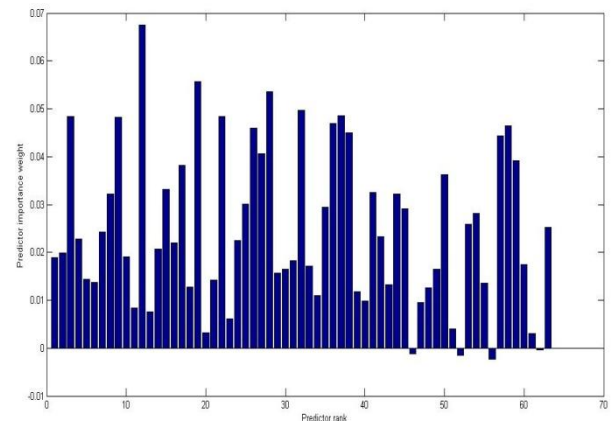
Thus the weight of any given feature decreases if it differs from that feature in nearby instances of the same class more than nearby instances of the other class, and increases in the reverse case. After m iterations, divide each element of the weight vector by m. This becomes the relevance vector. The important features for each cluster are given in table 3 and category wise the important features for each cluster are described in table 4.

#### 4. CLUSTER BASED RANKING (CBR) ALGORITHM FOR RESUMES

In order to rank the resumes and find the ranks the clustering results have been used. The algorithm is as follows:

- The feature vector with 63 words for all the resumes have been considered and the cluster value of the resume has been considered as the target value.

- ReliefF Algorithm has been used to calculate the weights to be assigned to each word count. The weights obtained are shown in the figure below:



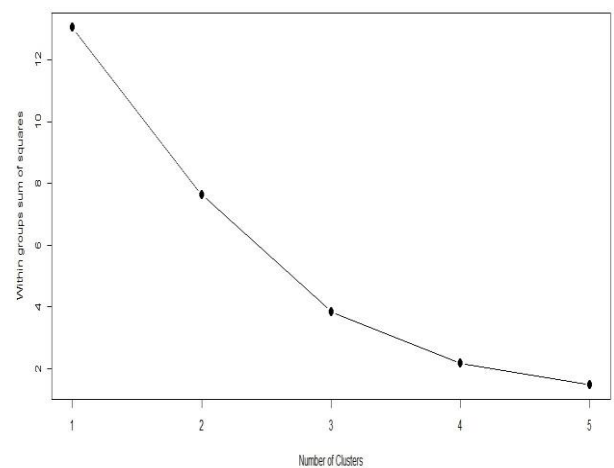
**Figure 3: Weights assigned to each word**

- Each word Count C(i) was then multiplied with the corresponding weight W(i) and all such products for a resume were summed to get the Cluster Based Ranking (CBR).

$$CBR = \sum_{i=1}^{63} C(i) * W(i), \text{ where } i \text{ is the number of words}$$

#### 5. RESULTS

In order to verify the ranking methodology, four test cases have been considered and the suitable candidates for those cases have been obtained. In the first case, ranking methodology has been applied on the Android Application Developers. The basic requirements which were taken into consideration were: Java, SQL, SDK or Visual Studio, XML, Android, Application and Develop. Using these basic requirements 12 candidates have been selected from all the available resumes. In order to rank these 12 candidates, clustering was performed.



**Figure 4(a): Determining number of Clusters Using Elbow Method**

Then the weights were calculated for calculating the CBR using the ReliefF and CBR was calculated.

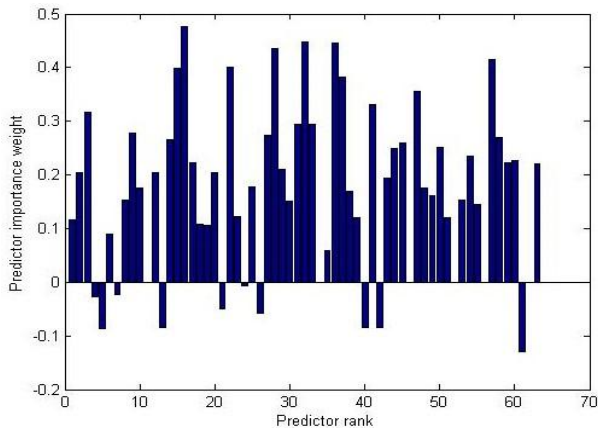


Figure 4(b): Weights assigned to each word

The top ranked resume which was obtained had following features:

- Around 4 years of UI development experience with HTML 4.0/5, CSS2/CSS3, XHTML, JavaScript, jQuery, JSON, Bootstrap, AJAX, Angular JS and PHP.
- Expertise in developing and updating a web page quickly and effectively using, HTML 5, CSS3, JavaScript and JQuery within the webpage cross browser compatibility.
- Expertise in creating the web page layouts using CSS and Twitter Bootstrap.
- Experience in using JavaScript libraries such as jQuery and GUI related Bootstrap etc.
- Knowledge in Cordova/PhoneGap to build mobile applications in IOS and Android.
- Proficient in using various IDE's like Dreamweaver, Sublime text editor 2/3, Notepad++, Atom, Visual Studio, and Netbeans etc.
- Strong exposure to Adobe tools - Photoshop, Dreamweaver and Prototype tools like Balsamiq.
- Used 960 Grid System to design templates in Photoshop.
- Knowledge in using Angular JS to create Single page application

For the second case Database Management candidates were selected. The basic requirements which were taken into consideration were: Database, Data, SQL, mySQL, Develop, Design, Scripting. Using these basic requirements 15 candidates have been selected from all the available resumes. In order to rank these 15 candidates, clustering was performed.

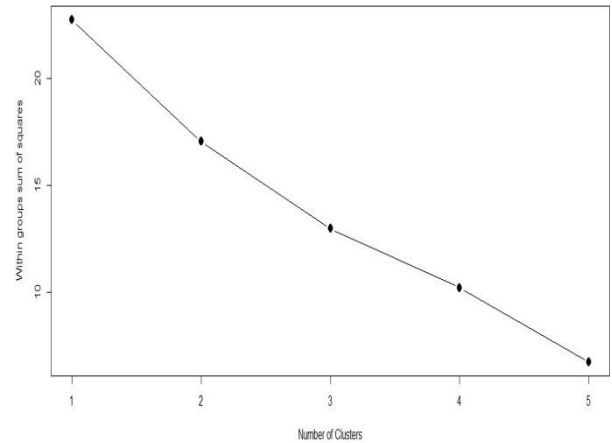


Figure 5(a): Assessing the number of clusters using Elbow Method

Then the weights were calculated for calculating the CBR using the ReliefF and CBR was calculated.

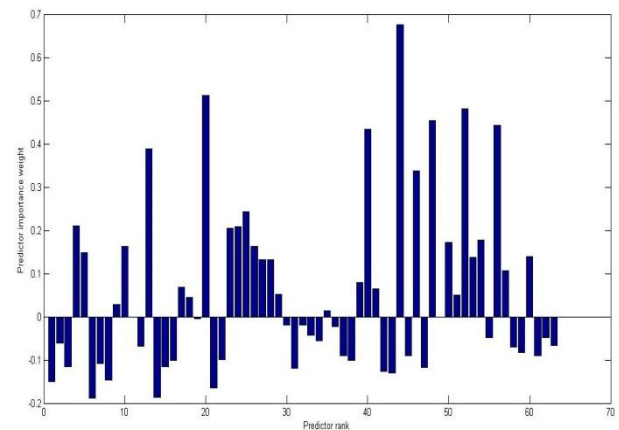


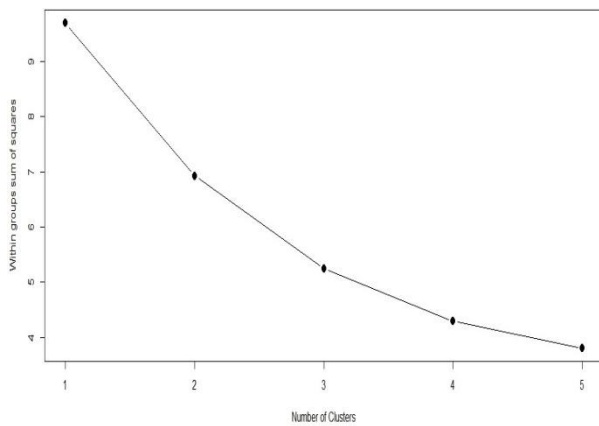
Figure 5(b): Weights assigned to each word

The top ranked resume which was obtained had following properties:

- Worked on Network based attendance system where he was responsible for client/server programming using sockets on SOLARIS. Created different tables, cursors, procedures, triggers in Oracle schema. Added Oracle data guard along with Setup of Oracle RAC facility for disaster management in case of hardware or other failure. Written UNIX shell scripts for cold and hot backup using crontab.
- Worked on Payroll Processing System where he created several shell scripts for payroll automation and wrote several SQL scripts to display the employee payroll information and updated the payroll information in database.
- Data Conversion and Loading . Written shell scripts using sed/awk for Extraction, Transformation and Loading of data using SQL. Loader from different heterogeneous source systems like Flat files, Excel, Oracle . Scheduled shell script for automatic run for backup/restore of different schema using exp/imp utility .Designed and developed UNIX scripts for creating, dropping tables which are used for scheduling the jobs .

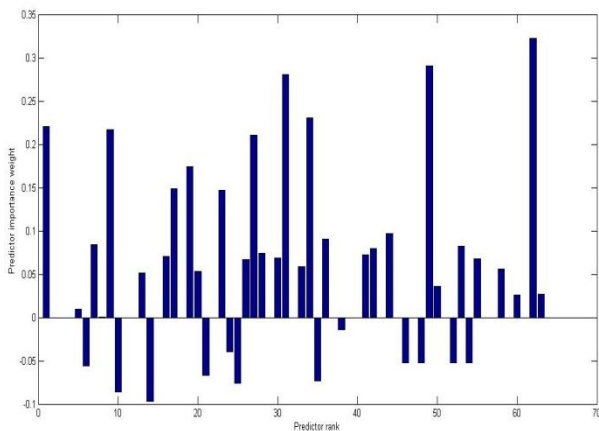
- Oracle Golden gate Live Replica for Data warehouse. Created Direct Load Configuration, initial load, Unidirectional and bidirectional. Configured the primary extract group.

For the third case IOS Developers were taken into consideration. The basic requirements which were taken into consideration were : iPhone, XCode, Develop, App, JSON, Data . Using these basic requirements 18 candidates have been selected from all the available resumes. In order to rank these 18 candidates, clustering was performed.



**Figure 6(a): Assessing number of clusters using the elbow method**

Then the weights were calculated for calculating the CBR using the ReliefF and CBR was calculated.



**Figure 6(b): Weights assigned to each word**

The top ranked resume had following features:

- Birth Control tales  
Deployment Target: iPhone.  
Front End Tool: Xamarin Studio  
Techniques Use: JSON Parsing, Xamrin.iOS, iOS Sdk  
Project Description:-This application made to help the user to get knowledge about controlling population and it's like a story book (Punchtantra) .
- E-Pub Reader  
Deployment Target: iPhone, iPad.  
Front End Tool: X-Code 8

Techniques Use: JSON Parsing, CoreData, Objective C, iOS Sdk

Project Description:-This application make the user to read book online, it also let's the user register and login. User can make bookmarks .

- Hotel Finder  
Deployment Target: iPhone.  
Front End Tool: X-Code 5

Techniques Use:JSON Parsing, Mapkit, Core Location, iOS Sdk

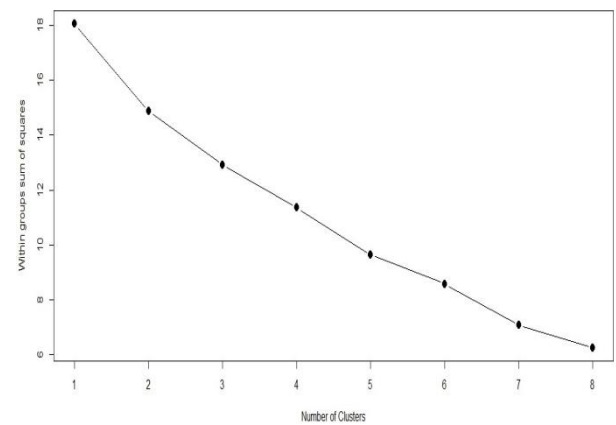
Project Description:-This application made to help the user to find out the location of Hotels. Shows the lowest and highest nightly rates of hotels. Shows location hotel on Google Map.

- Movie Scrawler  
Deployment Target: iPhone.  
Front End Tool: X-Code 6

Techniques Used: JSON Parsing, iOS SDK

Project Description:-This application made to help the user to find out the Movie release. Shows the description of the movie, it's released date, budget and popularity of the movie.

For the fourth case Web developers were considered. The basic requirements which were taken into consideration were: Web, Develop, html, CSS, javascript, jquery, php, website. Using these requirements 22 candidates have been selected from all the available resumes. In order to rank these 22 candidates, clustering was performed.



**Figure 7(a): Assessing number of clusters using the elbow method**

Then the weights were calculated for calculating the CBR using the ReliefF and CBR was calculated.

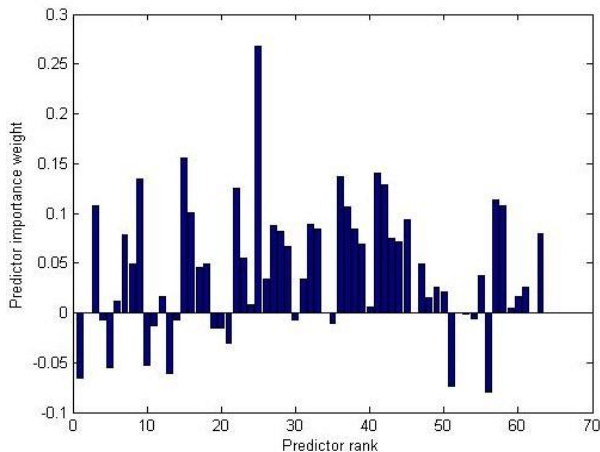


Figure 7(b): Weights assigned to each word

The top ranked resume had following features:

- Around 4 years of UI development experience with HTML 4.0/5, CSS2/CSS3, XHTML, JavaScript, jQuery, JSON, Bootstrap, AJAX, Angular JS and PHP.
- Expertise in developing and updating a web page quickly and effectively using, HTML 5, CSS3, JavaScript and JQuery within the webpage cross browser compatibility.
- Coded Java Script for page functionality and Pop up Screens.
- Experience in using JavaScript libraries such as jQuery and GUI related Bootstrap etc.
- Knowledge in Cordova/PhoneGap to build mobile applications in IOS and Android.
- Proficient in using various IDE's like Dreamweaver, Sublime text editor 2/3, Notepad++, Atom, Visual Studio, and Netbeans etc.
- Strong exposure to Adobe tools - Photoshop, Dreamweaver and Prototype tools like Balsamiq. Used 960 Grid System to design templates in Photoshop.
- Knowledge in using Angular JS to create Single page application.

## 6. CONCLUSIONS AND FUTURE WORK

The results clearly shows that the ranking methodology finds the suitable candidate for the desired job. This approach works very efficiently as shown by the results but since this will not be the final procedure that the company recruiter will follow because a personal interview of the candidate still holds its importance in the recruitment process. This approach will help the recruiters to find the relevant candidates which they can interview further and select the best candidate according to their choice. In this way the efforts put by the recruiters in going through every resume will reduce. In future the work can be done on how to effectively write the same resume through which the rank can be improved thus it will help the candidates to effectively showcase their talent. The work can be extended and evaluated on a large resume dataset to find the relevant candidates and clusters. Supervised

machine learning methods can be used based on the past history of selected candidates in order to rank the resumes and selecting the suitable candidate. Thus machine learning has a wide range of possibilities in this area and researchers have to explore them for making the recruitment process better.

## 7. REFERENCES

- [1] <https://www.linkedin.com>
- [2] <http://www.monsterindia.com/>
- [3] <https://www.naukri.com/>
- [4] <http://www.indeed.com/resumes>
- [5] Yu, Kun, Gang Guan, and Ming Zhou. "Resume information extraction with cascaded hybrid model." Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.
- [6] Chandola, Divyanshu, et al. "ONLINE RESUME PARSING SYSTEM USING TEXT ANALYTICS."
- [7] Kopparapu, Sunil Kumar. "Automatic extraction of usable information from unstructured resumes to aid search." Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on. Vol. 1. IEEE, 2010.
- [8] Zhi Xiang Jiang, Chuang Zhang, Bo Xiao, Zhiqing Lin, "Research and Implementation of Intelligent Chinese Resume Parsing", WRI International Conference on Communications and Mobile Computing, Jan 2009.
- [9] Zhang Chuang, Wu Ming, Li Chun Guang, Xiao Bo, "Resume Parser: Semi-structured Chinese Document Analysis", WRI World Congress on Computer Science and Information Engineering, April 2009.
- [10] Celik Duygu, Karakas Askyn, Bal Gulsen, Gultunca Cem, "Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs", IEEE 37th Annual Workshops on Computer Software and Applications Conference Workshops, July 2013.
- [11] Feldman, Ronen, and James Sanger. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, 2007.
- [12] Manning, Christopher D., and Hinrich Schütze. Foundations of statistical natural language processing. Vol. 999. Cambridge: MIT press, 1999.
- [13] <https://cran.r-project.org/web/packages/tm/tm.pdf>
- [14] [https://en.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier)
- [15] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.
- [16] Tibshirani, Robert, Guenther Walther, and Trevor Hastie. "Estimating the number of clusters in a data set via the gap statistic." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63.2 (2001): 411-423.
- [17] Liu, Huan, and Hiroshi Motoda, eds. Computational methods of feature selection. CRC Press, 2007.
- [18] [https://en.wikipedia.org/wiki/Euclidean\\_distance](https://en.wikipedia.org/wiki/Euclidean_distance)

## 8. APPENDIX

**Table 3: Top 15 Features of each Cluster**

Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7	Cluster8	Cluster9	Cluster10
API (0.060)	mySQL (-0.008)	Develop (0.056)	IPhone (-0.001)	Core (-0.025)	IPhone (-0.009)	Front (0.005)	Client (0.240)	Debug (0.032)	IPhone (-0.009)
Windows (0.00063)	Website (-0.016)	Plsql (0.048)	Xcode (0.002)	Eclipse (-0.0168)	Windows (-0.017)	Web (0.095)	AJAX (0.228)	Database (-0.018)	AJAX (0.114)
AJAX (0.106)	XML (0.047)	Software (0.036)	JSON (0.006)	Configurat ion (0.042)	Software (0.021)	Software (-0.014)	Automatio n (0.027)	App (0.001)	Analysis (0.064)
XCode (-0.015)	IPhone (0.029)	ADO.NET (-0.015)	Core (0.020)	Analysis (0.245)	Code (0.021)	Feature (0.130)	Database (0.126)	Json (0.028)	Client (0.102)
Plsql (-0.010)	Debug (0.028)	ASP.NET (0.001)	Javascript (0.013)	ADO.NET (0.062)	Html (0.007)	Configurat ion (0.020)	Php (-0.016)	CSS (0.324)	MVC (0.089)
Visual (-0.026)	Analysis (-0.0006)	Website (-0.0355)	Jqueri (0.009)	Server (0.139)	Servlet (0.158)	API (0.177)	Jqueri (0.340)	Scripting (-0.0078)	SQL (0.111)
Architect ure (-0.013)	Architectu re (0.005)	SQL (0.0998)	mySQL (0.014)	JSON (0.023)	Unix (0.025)	Android (0.173)	Scripting (0.172)	Core (-0.015)	Scripting (0.030)
Design (0.001)	Administra tor (-0.008)	Unix (0.153)	Database (0.007)	Jqueri (0.006)	Wordpres s (-0.0055)	ADO.NET (-0.019)	Plsql (0.026)	JQერი (0.228)	Shell (-0.0005)
IPhone (-0.009)	Android (0.070)	App (-0.022)	Administra tor (0.010)	API (0.041)	Design (-0.003)	Applicatio n (0.156)	Data (0.031)	Configurat ion (-0.017)	Code (0.108)
Sqlite (-0.011)	Descript (0.064)	Descript (0.018)	Servlet (0.015)	Applicatio n (0.181)	Architect ure (0.042)	SDK (0.261)	Administra tor (0.172)	Website (0.213)	Bootstrap (0.020)
Javascript (0.084)	JSP (-0.008)	JSP (-0.0063)	XML (0.009)	Code (0.154)	Plsql (0.010)	SQL (-0.020)	Backend (0.164)	Network (0.015)	Debug (0.032)
Java (0.002)	Wordpress (-0.008)	Intern (0.040)	Network (0.009)	Architectu re (0.169)	XML (0.022)	XCode (-0.020)	Architectu re (0.364)	Develop (0.040)	Administra tor (0.0297)
Network (0.020)	Develop (0.020)	Automati on (0.069)	Intern (0.008)	Php (-0.265)	Backend (0.017)	Network (0.282)	MVC (0.270)	Software (-0.028)	Jqueri (0.078)
Backend (0.043)	JSON (0.077)	JSON (-0.036)	MVC (0.0008)	SDK (-0.0055)	Descript (0.028)	Sqlite (0.213)	Core (0.004)	ASP (-0.0065)	Database (0.049)
Applicati on (-0.001)	Linux (0.001)	Server (0.122)	Html (0.027)	Unix (0.022)	Intern (0.031)	Design (0.085)	Module (0.150)	Analyst (0.022)	Javascript (0.080)

**Table 4: Category wise top fifteen words in all clusters**

Category	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9	Cluster 10
Role	Architect ure ,Design, Network, Backen d, Applicat ion	Debug, Analysis, Architect ure, Administ rator, Descript, Develop,	Develop , Descript , Intern, Server	Database, Administ rator, Network, Intern,	Analysis, Server, Applicati on, Architect ure,	Design, Architec ture, Backend , Descript , Intern,	Front, Web, Applicati on, Network, Design,	Client, Database, Administ rator, Backend, Architect ure,	Debug, Database, Network, Develop, Analyst	Analysis, Client, Debug, Administ rator, Database
Language, Databas e and Web related skills	Plsql, sqlite, Javascr ipt, Java	mySQL, XML, JSON	Plsql, ADO.NET, ASP.NET, SQL, JSON	JSON, Javascript , Jqueri, mySQL, Servlet, XML, HTML	ADO.NET, JSON, Jquery, Php,	HTML, Servlet, plsql, XML,	ADO.NET, SQL, Sqlite,	Php, Jqueri, Plsql,	JSON, CSS, Jqueri,	SQL, Shell, JQერი, javascript ,

Package s, Tools and Frame works	AJAX, XCode, Visual,	JSP, Wordpre ss,	JSP	XCode, MVC,	Eclipse, SDK,	Wordpre ss,	SDK, XCode	AJAX, MVC	ASP	AJAX, MVC, Bootstrap ,
Experi ence Skill mention ed in the resume	API	Website	Softwar e, Website , App, Automa tion,	Core,	Core, Configur ation, API, Code,	Software , Code,	Software, Feature, Configur ation, API,	Automati on, Scripting, Data, Core, Module,	App, Scripting, Core, Configur ation, Website, Software,	Scripting, Code,
OS	Window s, IPhone	IPhone, Android, Linux	Unix	IPhone,	Unix	IPhone, Window s, Unix,	Android			IPhone,