International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 11, November-2016

Proposed System for Resume Analytics

Amala Deshpande
Department of Computer Engineering,
VESIT.
Mumbai, Maharashtra 400074

Divya Deshpande
Department of Computer Engineering,
VESIT.
Mumbai, Maharashtra 400074

Deepika Khatri
Department of Computer Engineering
VESIT.
Mumbai, Maharashtra 400074

Prarthita Das
Department of Computer Engineering,
VESIT
Mumbai, Maharashtra 400074

Faculty Mentor
Sujata Khedkar
Department of Computer Engineering,
VESIT
Mumbai, Maharashtra 400074

***Abstract*— This paper aims at proposing an automated system to shortlist the best résumés and make it easier for the human resources department to select candidates. The human resources department only has to upload the résumés, which would be normalised and clustered according to various parameters. The clustered résumés are then scored based on the criteria specified by the HR department and sorted in decreasing order of their scores.**

***Keywords*—K-means clustering, normalization, web crawler, entity extraction, resume**

INTRODUCTION

Technology today has made it possible to send a résumé within the tap of a button. Hundreds of résumés are being received for a particular job posting. This makes the job of an HR department especially difficult as it is impossible to peruse through each of the résumés and shortlist the candidates.

Moreover, each résumé has a different format. There is a need to extract the relevant information from the resume and store it in the database, so that sifting through the data becomes easier. In order to make the job of a recruiter easier, we propose an automated system that does most of the preliminary filtering and presents the data in a standard format.

I. RELATED WORK

Information extraction plays an important role in resume analytics as the unstructured resumes need to be normalised into a standardised format for further processing. Previously, systems have been designed that extract several important informative fields from resume using natural language processing techniques. These systems are web based client-server which is capable of extracting information from resumes in English language[1]. Other systems use methods like pattern matching and computing the term frequency by following a set of patterns. A recursive algorithm is applied to determine frequent compound patterns[2].

In the case of résumés, each résumé is a human-made text and has the freedom of choosing the words, format, structure, and content. To make a comparison between varying documents, a platform has to be provided to bring all documents on the same scale. The resumes need to be normalized in order to map them to an existing database or to compare their values[9]. To facilitate a better normalization of data, techniques like Named Entity Normalization (NEN) need to be applied to the content to detect and resolve similarity between two entities[7]. A proposed system generates a database constructed by parsing and altering these mappings, and indexing the mappings for quick access and matching operations[7]. Another study proposes a system called Carotene which incorporates SVM as a coarse level classifier. SVMs are robust on sparse and high-dimensional data such as job title data sets. Carotene further uses kNN proximity based vertical classifier as a fine level classifier[8]

Various different clustering approaches have also been studied earlier, some which use strict clustering technique to group the resumes into exactly one cluster. The cleaned, filtered, converted and extracted data from the resumes are clustered according to various parameters enabling the recruiter to discover the exact matches of candidates he/she needs. The relevancy ratios are also computed which serve as a parameter for checking how relevant a resume is as compared to all the resumes present in the dataset [4]. Class overlapping is a problem associated with clustering, which is a result of ambiguity in placing a resume in a given cluster as it matches more than one. To overcome this, many schemes are used for finding and dealing with the class overlapping problem, which include schemes like discarding schemes, merging schemes and separating schemes [3].

The process of filtering resumes is mainly based on comparing the candidate data with the job requirements. This process gives all the candidates who match the description. To make the process more efficient, a score is given to each resume to rank the candidates. However, owing to the large number of resumes the candidate scores have less dispersion. The technique of collaborative filtering is used to adjust the scores and improve score quality [6]. Collaborative filtering is a technique that can be used to predict the trend of selection [6]. Another factor which is considered in certain proposed systems is the risk factor after recruitment [5]. Associate rule mining technique is applied to patterns in historical data of the organization which satisfy minimum support and confidence and then final rules are framed [5]. The system proposed in [5] first applies prerequisite rules provided by recruiter to the candidate profile and then associate mining rules are applied.

II. PROPOSED SYSTEM

The system proposed downloads resumes using a web crawler. The downloaded resumes are unstructured data set in .pdf format. These are converted to .json format using information extractor format. The converted resumes are normalized to bring all resumes on the same platform. The resumes are further clustered based on skills, education, work experience. The system proposed is made to help recruiters get the best candidates for a particular job profile. To facilitate this dynamic clustering is performed and resumes are scored based on unit scoring method. The Figure 1 shows the module diagram of the proposed system.

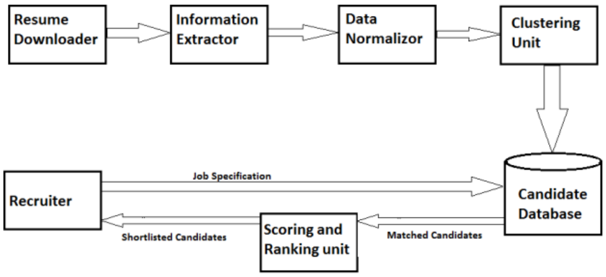

Figure 1. System Design

A. Information Extraction

Information Extraction (IE) is a kind of Information Retrieval method used to automatically extract structured information from a large collection of unstructured documents. The downloaded resumes are parsed using the Resume Parser we have designed. The resume parser is semantic in nature.

B. Normalization

Normalization or rescaling is performed to translate values in different ranges to the same scale. We perform the normalization on the strings extracted from résumés by comparing canonical names from the database.

C. Clustering

Clustering can be defined as the process of creating clusters. Each cluster is a collection of objects which are similar in some manner. It usually deals with finding a similarity in an unstructured collection of unlabeled data. In this system the clustering is done based on skills and work experience.

D. Scoring and Ranking

Each recruiter can have different set of specifications for a particular job title. The system aims to provide candidates whose qualifications and skills match the recruiters' specifications. The previous units cluster the candidates based on skills and work experience. This unit first checks for the job specifications provided by the recruiter. Then parameters are decided based on which second clustering has to be done. Further each candidate is scored to generate a final ranking of the best matched candidates.

III. INFORMATION EXTRACTION MODULE

The algorithm demarcates the file according to various headings like Name, Age, Address, Job Title, Educational Qualifications, Work Experience and technical skills. After demarcation, various threads under the same headings are run simultaneously. Each thread parses through the already made demarcations and extracts the information from the body. The extracted information is then stored into a json file under the same heading as the thread. Once all the threads have executed, the json file is ready and we have extracted the relevant data from the raw pdf files.

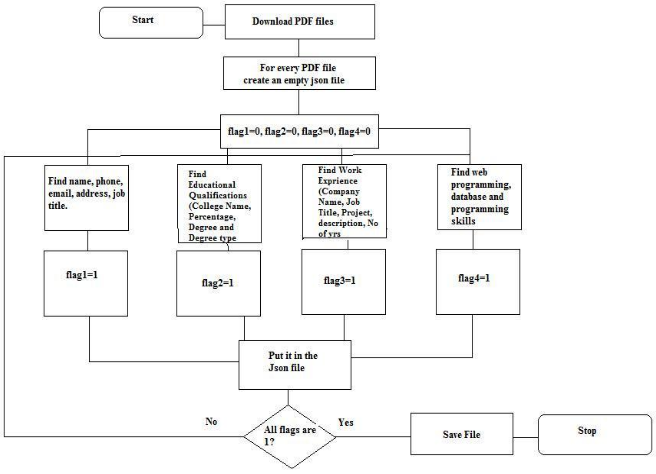
Start
Download PDF files
For every PDF file create an empty json file
flag1=0, flag2=0, flag3=0, flag4=0
Find name, phone, email, address, job title.
Find Educational Qualifications (College Name, Percentage, Degree and Degree type
Find Work Exprience (Company Name, Job Title, Project, description, No of yrs
Find web programming, database and programming skills
flag1=1
flag2=1
flag3=1
flag4=1
Put it in the Json file
No
All flags are 1?
Yes
Save File
Stop

Figure 2. Information extraction Flowchart

IV. CLUSTERING UNIT

This unit classifies all resumes in different clusters. The algorithm for the clustering process is as follows-

1. The clustering algorithm used is K-Means Clustering algorithm and an hierarchical approach is adopted.
2. In the first stage, the normalised resumes are passed on as inputs and it is clustered based on the years of work experience acquired by each applicant.
3. Upon deciding the number of clusters, K-Means algorithm is used to compute the centroid and minimum distances between the objects.
4. Once the objects are determined of each cluster, we ultimately gain the clusters depending on years of work experience, which can further be evaluated.
5. All this is depicted in Fig. 3
6. In Fig 4, the second phase of hierarchical clustering is depicted in which the now clustered resumes are further clustered on the basis of their skill sets.

7. This clustering is done, by passing the previous cluster sas the input and applying K-Means algorithm to them again.
8. This clustering is done on the basis of skill sets, which are further classified into three types:

- Programming languages (eg. java, Python)
- Database tools (eg. MySQL, Oracle)
- Web designing tools (eg. HTML, CSS, PHP)

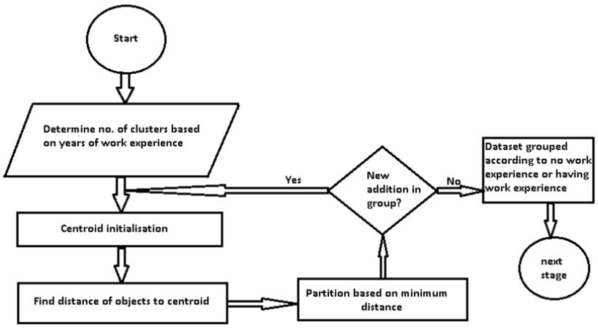
Start
Determine no. of clusters based on years of work experience
Centroid initialisation
Find distance of objects to centroid
Partition based on minimum distance
New addition in group?
Yes
No
Dataset grouped according to no work experience or having work experience
next stage

Figure 3. Phase 1 of clustering

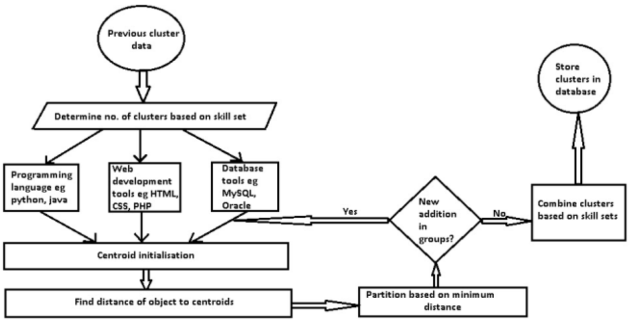
Previous cluster data
Determine no. of clusters based on skill set
Programming language eg python, java
Web development tools eg HTML, CSS, PHP
Database tools eg MySQL, Oracle
Centroid initialisation
Find distance of object to centroids
Partition based on minimum distance
New addition in groups?
Yes
No
Combine clusters based on skill sets
Store clusters in database

Figure 4. Phase 2 of clustering

V. SCORING AND RANKING

This module performs dynamic classification based on recruiter's job specification, and displays top N candidates ranked on the basis of a score given to each candidate. The algorithm for this module is as follows-

1. Recruiter provides no of candidates he wants to call for interview and job specifications for the job title (like skills required, education, no of years of work experience)
2. This module checks whether the job specifications given by the recruiter are already present in previous stage clusters, if not present they are added to new cluster list
3. If present then those clusters are marked.
4. A final candidate list is generated for a particular recruiter by clustering based on marked clusters and new cluster list made.
5. The final candidates are scored by following rules-
6. Work experience score is equal to no of years of relevant work experience
7. Skills set is given score 1 each, but skills used in a project are given score 2.
8. Education of engineering in CS or IT is given score 2 and rest are given score 1, if candidate has master's then score is incremented.
9. The final score is calculated by adding the 6,7,8
10. The candidates are arranged in descending order of score and top N candidates are displayed to the recruiter.

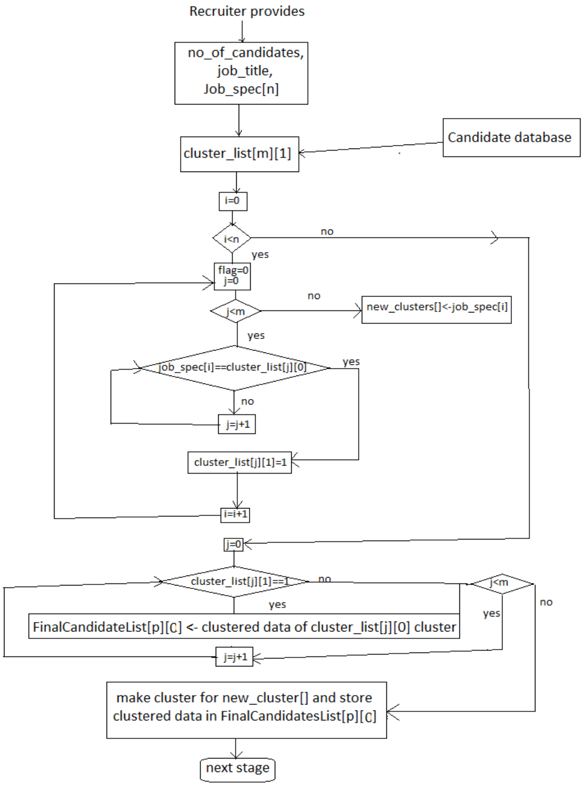
Recruiter provides
no_of_candidates,
job_title,
Job_spec[n]
Candidate database
cluster_list[m][1]
i=0
i<n
no
yes
flag=0
j=0
j<m
no
new_clusters[]<-job_spec[i]
yes
job_spec[i]==cluster_list[j][0]
yes
no
j=j+1
cluster_list[j][1]=1
i=i+1
j=0
cluster_list[j][1]==1
no
j<m
yes
no
yes
FinalCandidateList[p][C] <- clustered data of cluster_list[j][0] cluster
j=j+1
make cluster for new_cluster[] and store clustered data in FinalCandidatesList[p][C]
next stage

Figure 5. Dynamic Classification

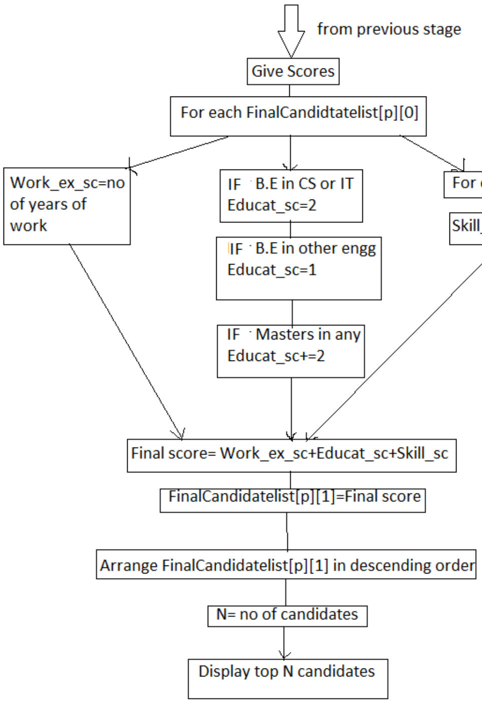
from previous stage
Give Scores
For each FinalCandidtatelist[p][0]
Work_ex_sc=no of years of work
IF B.E in CS or IT Educat_sc=2
IF B.E in other engg Educat_sc=1
IF Masters in any Educat_sc+=2
For each skill
Skill_sc+=1
Final score= Work_ex_sc+Educat_sc+Skill_sc
FinalCandidatelist[p][1]=Final score
Arrange FinalCandidatelist[p][1] in descending order
N= no of candidates
Display top N candidates

Figure 6. Scoring Algorithm

VI. CONCLUSION

We have proposed this system to make it easier for the recruiter to select candidates. It also presents the information in a standardised format. The raw data we acquired through the résumés is normalised, clustered and scored to display the top N candidates. We have also incorporated the recruiters demands while scoring the resume, thus making it recruiter specific.

VII. FUTURE SCOPE

Further advancements that can be done with our proposed approach is:

1. Scoring can be done based on weights given to each parameters. Higher weights can be given to more relevant parameters. The relevancy of the parameters can be measured using past recruitment trends.
2. Personality analysis can be done of the shortlisted candidates using social media information provided in the resumes. This analysis will help to judge whether the candidate's personality as per his/her social life matches the job requirements.

REFERENCES

[1] Sunil Kumar Kopparapu, "Automatic Extraction of Usable Information from Unstructured Resumes to Aid Search," published in Progress in Informatics and Computing (PIC), 2010 IEEE International Conference

[2] V. Jayaraj, V. Mahalakshmi, P. Rajadurai, "Resume Information Extraction using Feature Extraction Model" published in American International Journal of Research in Science, Technology, Engineering & Mathematics, June-August, 2015.

[3] Haitao Xiong and Junjie Wu Lu Liu, "Classification with class overlapping: A systematic study," in 2010 International Conference on E-business Intelligence.

[4] V. Jayaraj and P. Rajadurai, "Information extraction using clustering of resume entities," published in 01 January 2016 publication in International Journal of Science Technology and Management.

[5] Dr Lakshmi Rajamani, Mohd Mahmood Ali, "Automation of decision making process for selection of talented manpower considering risk factor: A Data Mining Approach", published in IEEE 2012

[6] Chanawee Chanavaltada, Panpaporn Likitphanitkul, Manop Phankokkraud, "An Improvement of Recommender System to Find Appropriate Candidate for Recruitment with Collaborative Filtering", published in 2015 ICCSS

[7] Ferosh Jacob, Faizan Javed, Meng Zhao, Matt Mcnair, "sCooL : A System for Academic Institution Name Normalization", published in IEEE 2014

[8] Faizan Javed, Qinlong Luo, Matt McNair, Ferosh Jacob, Meng Zhao, Tae Seung Kang, "Carotene: A Job Title Classification System for the Online Recruitment Domain", published in 2015 IEEE First International Conference on Big Data Computing Service and Applications

[9] Charul Saxena, "Enhancing Productivity of Recruitment Process Using Data mining & Text Mining Tools", San Jose State University