Automated Filtering of Relevant Resumes

Prarthita Das
V.E.S.I.T
Mumbai University

Amala Deshpande
V.E.S.I.T
Mumbai University

ABSTRACT
In today's world, there is a colossal rise in the number of job seekers. Additionally, with increasing applicants the number of resumes increases proportionally, thereby making the task of the HR department extremely laborious. Therefore, a new approach has been proposed to automate the task of recruiting suitable candidates for a given job profile by matching the candidate's qualities to the parameters mentioned by the recruiter which involves parsing the resume automatically. This system, takes the unstructured or structured resume as input from the applicant and the job specifications from the recruiter (which acts like a query) and then using information extraction, storing and matching techniques a certain relevancy percentage is calculated which determines the extent to which the candidate is suitable for that post. The higher the percentage, the better the candidate is for that portfolio, thereby providing the recruiter with the best results for that given job profile. Therefore, this proposed system makes the task of recruiting more efficient and faster, and also eliminates the need to manually find the best suited applicants.

General Terms
Resume analytics, Ranking system, Information extraction

Keywords
Relevancy. Collaborative filtering, unstructured, resume, matching

1. INTRODUCTION
Large enterprises and organizations receive millions of resumes by prospective job applicants regularly. Since there is no standard format in which a resume can be written, the resumes received might be of varying formats (e.g. .txt, .pdf, .docx) which makes it difficult for companies and organizations to analyze and select the most ideal ones out of the hordes of resumes that they receive. Initially, different approaches were adopted in order to parse resumes which included applicants filling out a predefined template which varied from organization to organization as per the job requirements. Although, using a predefined template, makes the process of extracting relevant information easier yet it limits the domain of information that can be extracted as the requirements change with every job description and organization. Therefore, better methods were developed for extracting information from unstructured data as present in the resumes and then storing the extracted content in an electronic database. The extraction of relevant information is based on a set of natural language processing and pattern matching techniques [1]. Furthermore, methods were developed in order to extract the promising features from the text, by adopting the feature selection process which is carried out to filter out the unwanted meaningless text from the data content of the resumes. This new method reduces the space and the dimension of the text considerably. When a recruiter needs some resumes, he/she performs a keyword search on this resume database. This search uses dynamic filtering techniques. The set of filters help in tossing out irrelevant resumes [2].

Ideally, the extracted data needs to be categorized into clusters for faster computation. Various different clustering approaches have been studied, some of which use strict clustering techniques 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 match 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 [3]. 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 [4].

The process of filtering resumes is mainly based on comparing the candidate data with 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 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 [5]. Collaborative filtering is a technique that can be used to predict the trend of selection [5]. Another factor which is considered in certain proposed systems is the risk factor after recruitment [6]. 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 [6]. The system proposed in [4] first applies prerequisite rules provided by recruiter to the candidate profile and then associate mining rules are applied.

In this paper, a better approach is discussed in order to extract the resumes considering the requirements of both the applicant and the recruiter in natural language. In order to select the most ideal candidates for the given job portfolio, ranking of the resumes would be done by giving a certain relevancy percentage to the shortlisted resumes which would determine the top ranks of the results as the output to the recruiter. These finalized resumes, would be the most ideal fit for the given job description thereby making the process of recruiting for the HR department easier.

2. SYSTEM DESCRIPTION
This is a web-based system which is capable of automatically extracting information from resumes in .doc or .pdf format and storing them in NoSQL type database. The job recruiter query is matched to resume database to get the most relevant candidates. Further Relevancy percentage and Collaborative filtering is used to score and rank the resumes. It will include various modules for carrying out the task of seeking the ideal resumes, like the information extraction module, the database

module and also a module for scoring the selected and processed resumes which will ultimately help in ranking them accordingly.

The interface will have tabs for the recruiter as well as the job seekers. The applicant will make his account on this online portal and upload his resume in the .pdf or .doc format. This uploaded PDF will get stored into the database built and is converted to .json format. The .json file now created will be loaded into the MongoDB database.

The database now builds in MongoDB after many job seekers apply. The recruiter's query is taken as a text input on the portal. The query will be regarding the job specifications needed by the recruiter for a particular job profile. After the recruiter's query is given matching is performed to find out the extent to which the requirements of the recruiter match with the stored resumes. The resumes are scored based on relevancy percentage and collaborative filtering. Furthermore, the shortlisted resumes are now ranked according to their scores and given as the outcome to the recruiter in descending order of precedence, thus giving the recruiter the highest ranked resume and then the following top 50 results.

The figure below demonstrates the design for the proposed system.

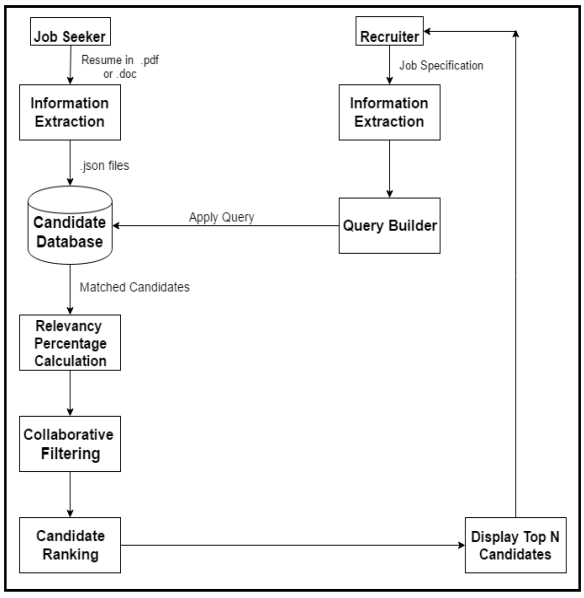

Figure 1: System Design

The modules used by this system are further explained in detail in the following sections.

2.1 Information Extraction

This module takes the resume in .pdf or .doc format as input and gives the output in .json format. Here, a code in Java language using *Apache Tika framework* parses the resume and extracts fields like name, skills, education, work experience and miscellaneous. A .json format file is created having the same fields. Dictionaries are used to match the field headings of json file with the resume. For example, the resume can contain skills under various headings like *"Skills", "Skill set", "Technical skills"*, etc. So, to correctly extract the skills a dictionary containing all variations need to be used. Similar is the case with all other fields. There is also a provision to extract skills used in a particular job experience or projects. This facility will enable the matching of skills implemented in real world projects. For the candidate profile .json type of file format is chosen, as it supports dynamic schemas, hence our algorithm has the provision to add any other fields given in the resume.

John Thomas

London, U.K.

Work Experience

Python Developer
ABC solutions limited - London
July 2015 to Present

Skills Used
Java Servlets, Python, RabbitMQ, Redis, twisted-python, django, scaling, performance.

Education

B.E. in Electronics Engineering

Skills

Java, Servlet, Python, RabbitMQ, Redis, Docker, twisted, Nodejs, html, CSS, js, ajax json (Less than 1 year)

Additional Information

Easily adaptable.
Focused on the task.
Eagerness to learn.

Figure 2: Resume in .pdf format

The Figure 2 shows the unstructured resume of a candidate as uploaded on the web portal. The fields from this unstructured resume need to be extracted to derive useful information from it.

The Figure 3 shows the .json format of the resume with the extracted fields. Now this format can be stored in a NoSQL database.


```
{
   "name":{
      "first":"John",
      "middle":"",
      "last":"Thomas"
   },
   "address":[
{"city":"London",
"country":"U.K"
}],
   "work_experience":[
      {
         "date_start":"July 2015",
         "jobtitle":"Python developer",
         "organization":"ABC solutions limited ",
         "date_end":"October 2016",
         "duration":"1 year 3 months"
      } ],
   "skills":[
      {
         "skills":"Java, Servlet, Python, RabbitMQ, Redis,
Docker, twisted, Nodejs, html, CSS, js, ajax json"
      }
{        "skills used":"Java Servlets, Python, RabbitMQ, Redis,
twisted-python, django, scaling, performance"
      }  ],
   "education_and_training":[
      {
         "Education":"B.E. in Electronics Engineering"
      }],
   "misc":[
      {  "Additional Information":"Easily adaptable,
Focused on the task,Eagerness to learn"}]}
```


Figure 3: Candidate Profile in .json format

2.2 Storing in database and Matching

The .json format files are stored in MongoDB database. The MongoDB database is a free and open sourced database platform. Being, classified as a NoSQL database program, MongoDB avoids relational database structure in favor of JSON-like documents making the integration of data simple in our project. In this module, the recruiter provides the job specification in text format, which is converted to a MongoDb query. The query is then given to the Candidate database to find the candidates that best suit the job requirements as predefined by the recruiter.

The MongoDB recruiter query is now used to find the candidates whose profile matches any of the parameters mentioned by recruiter, by simply matching the content of the query with that of the extracted database content.

2.3 Relevancy Percentage

Relevancy Percentage is the percentage which is calculated depending on the extent to which the qualities of a given candidate matches with the parameter mentioned by the recruiter. Now, a large number of candidates will be having at least one parameter matching to the recruiter's specification. Hence, Relevancy Percentage is calculated for each field. For instance, if the recruiter enlists 10 skills required for a particular job and a candidate has 8 of the mentioned skills then his relevancy percentage for that field will be 80%. Likewise, Relevancy Percentage is calculated for all fields and averaged to get a final percentage for the candidate. This process ensures elimination of least relevant candidates. The relevancy percentage is calculated for all the matched candidates. Candidates below 50 to 60% relevancy are eliminated.

2.4 Ranking

Though most of the candidates get eliminated based on relevancy percentage, many of the remaining candidates might still have the same relevancy percentage. Thus, it becomes difficult to distinguish between these candidates having the same percentage. Therefore, in this scenario collaborative filtering technique can be used to disperse the relevancy percentage. Collaborative filtering is a technique used by recommender systems to predict interests. In this system, it can be used to predict the tendency of the recruiter in case of a candidate with a particular qualification and skill set [5]. Here, the past records of candidates getting recruited through the system can be used for serving the purpose of collaborative filtering. The skill set of the candidates who were recruited by companies in the past is matched with all the relevant candidates and an additional score is given to each candidates based on the matching percentage. Finally, an overall score is given to each candidate based on collaborative filtering and relevancy percentage. The candidates are then ranked according to the final score and the top 50(or depending on recruiter's demand) candidates are given as the output to the recruiter. The candidates chosen through this system are the most ideal ones and this also saves the recruiter's efforts of manually scanning the resumes.

3. FUTURE SCOPE

The systems designed so far extracts all the information about the candidate only through his/her resume and after extraction it stores the information in a centralized database, finally ranking them and giving the top 50 results to the HR recruiter according to their specifications. Future advancements in this system can be as follows:

1. The profiles of the candidates can be tracked on social media sites as this will help in analyzing the personality of the candidate and whether he/she is a perfect fit for the post can be judged
2. Analysis can be done over the past records of the candidate which will help us determine his expected tenure of work in the organization.

4. CONCLUSION

This work has made an extensive effort to provide a system through which the resumes can be ranked with maximum efficiency. By implementing this system, the task of obtaining the most relevant resumes can be achieved which will save the recruiter time to manually select appropriate resumes, which even after processing may not be a complete fit for the profile. Since, there are multiple levels of screening involved in order to find the most relevant resumes; the accuracy of the system also improves. Thus, using this ranking technique, we can obtain the best results for obtaining the ideal resumes. This approach will automate as well as speedup the process of the HR recruiters.

5. REFERENCES

[1] Sunil Kumar Kopparapu, "Automatic Extraction of Usable Information from Unstructured Resumes to Aid Search", published in IEEE 2010

[2] V. jayaraj, V. Mahalakshmi, P. Rajadurai, "Resume Information Extraction using Feature Extraction Model", published in AIJRSTEM 2015

[3] 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.

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

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

[6] 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