# An Unstructured Text Analytics Approach for Qualitative Evaluation of Resumes

Vinaya R. Kudatarkar*
*Department of Computer Science*
*KLS Gogte Institute of Technology*
*Belagavi, Karnataka, India*

Manjula Ramannavar
*Department of Computer Science*
*KLS Gogte Institute of Technology*
*Belagavi, Karnataka, India*

Dr.Nandini S. Sidnal
*Department of Computer Science*
*KLE College of Engg. & Tech.*
*Belagavi, Karnataka, India*

*Affiliated to Visveswaraya Technological University*

***Abstract*— With the growing use of more and more data on networks, big data has become the new trend for productivity, innovation and competition across companies and industries. The proliferation of textual data in businesses is overwhelming. Unstructured or semi-structured textual data is being constantly generated via web logs, emails, documents on the web, blogs, and so on. While the amount of textual data is increasing very rapidly, there is demand for the ability to summarize and analyze in order to make sense of such data for making good business decisions This work reviews how to organize and understand the textual data and presents an unstructured text analytics approach for qualitative evaluation of CV/Resume documents. An effective approach for extracting the resume information from websites and analyzing it thereby making the job easier for finding suitable resumes is presented. The results obtained are a fair measure of qualitative account of a resume document on the parameters of coverage, and comprehensibility and demonstrates the usefulness of the proposed algorithmic approach.***

***Keywords*— Big data; Concept Extraction; Qualitative evaluation; Text Analytics; Resume parsing**

## I. INTRODUCTION

The escalation of textual data in business is massive. unstructured or semi-structured textual data [22] is being constantly generated via web logs, emails, documents on the web, blogs, and so on. while the amount of textual data is increasing very rapidly, there is demand for the ability to summarize, analyze and make sense of such data for making good business decisions. for companies and industries, it is very challenging to understand and analyze unstructured textual data. text analytics thus becomes the key to solving enormous business concerns."

Text analytics is a qualitative approach which is faster, easier and highlights important concepts hidden in the unstructured or semi-structured textual data. text analytics/text mining [32] refers to deriving high quality text from unstructured or semi-structured documents. the quality of text is derived from a combination of interestingness measures, relevance and originality or from trends and patterns. choosing the appropriate key words for search is one of the crucial aspects for getting the desired results. it is a very interesting fact that the internet generates lot of qualitative textual data through numerous conversations. thus improving the precision of search is the most vital functionality in industries and companies like yahoo and google. many technologies have begun to fill the "gap between" computer and human language. the processing of natural language has come up with many technologies where computers understand, analyze and come up with new text. the technologies which have been developed are information extraction, concept linkage, categorization, summarization, clustering, topic tracking, question answering, information visualization and association analysis. data mining [32] is a related technology that handles only structured text from databases. on the other hand, text analytics can handle both semi-structured and unstructured data sets.

The rest of the paper is organized as follows: section 2 provides related work and discusses various types of resume parsers. section 3 presents system design and section 4 describes implementation details. section 5 illustrates experimental results of the proposed approach. section 6 summarizes the paper in the form of conclusion. section 7 provides directions for future work.

## II. RELATED WORK

A career for graduates is all about getting placed in a good organization where their rights are protected and benefits are looked after. Recruiters look for individuals who would contribute effectively for organizational growth. The importance of a resume is that it acts as the first impression of a candidate. First impression count is a measure for the employer to make a decision to eliminate or retain the candidate for subsequent rounds of the recruitment process. Reasonable usage of Power Words makes the sentences of a resume stronger. Strong resumes can help you to outstand and pursue to the next round of recruitment. There are many types of resumes such as, chronological resumes, visual resumes, functional resumes etc.

Statistics indicate that an employer hardly spends a lot of time in looking at the resumes. In this competitive world, a candidate should possess a powerful resume that conveys the required information in a manner that it stands out among resumes of contemporaries. The databases of companies hold lakhs of resumes which are unstructured and in free style. The information and the structure contents of resumes will be collection under sub topics; the classification and the representation of information differ from one another.

Hence gathering relevant data from each resume and storing it into the companies' database in a particular format would reduce human effort. There are some difficulties of resume service by unions or commercial companies since they consume too much of time, capacity, money, human effort and so on. These companies require filtered/parsed resumes for the recruitment process. Automated recruitment systems require that Job seekers post their resumes on various websites like Indeed.com, LinkedIn, Naukri.com, Monster.com, Resume builder etc. Certain websites may retrieve unwanted resumes while some may provides very minimum number of resumes. This calls for an approach for qualitative evaluation of resumes.

### *A. Resume Parsing*

Resume parsing is the process of analyzing the document and extracting elements or concepts of the resume of what the writer actually meant to say like, his skills, education, experience, achievements and so on. Resume parsing is also known as CV Extraction, CV Parsing or Resume Extraction. Resume parsing is a tool which captures all different ways of writing resumes through complex algorithms and complex rules.

Related work can be placed into 8 categories.

#### *1) Parsing based Keyword search*

The authors in [20] introduced the issues of keyword search based on extracting; that is, given the syntax, a database of information and queries, the goal was to return back most clearly defined parse. It was shown that this issue was hard to deal with; they found that the issue of returning all apropos parses was highly complex. Eventually, they rewrote rules and showed that under some certain condition, finding out all the rewritten extracts is computable. The work focuses on productivity, that is how to design index structures and algorithms productively to support a parsing based search of keywords. The work differs from previous works in the use of a scoring task to express parse preferences. There are two advantages to the scoring task approach compared to the containment approach:

(1) Containment based ordering is shown to be NP-hard, while scoring task ordering is manageable for various interesting classes

(2) Scoring task urges a total ordering over the parses unlike containment.

#### *2) Dictionary based Segmentation*

In [21] the problem of string segmentation while sharing search for the structured document was considered. However, earlier work did not parse with grammar. The work uses systematic step by step procedure to improve productivity. Another previous work concentrates on query segmentation using Conditional Random Fields (CRFs). A notion similar to maximal matches was used to establish matches that also appear in the database. However this work does not support syntax and relational constraints. There also been some work done on web queries that does not support syntax and relational constraints too. It further restricts the matches to be keys from a single table. Nevertheless, some of the techniques may be used to infer patterns.

#### *3) General keyword search*

There has been much work done on keyword search [22] in databases over the last many years. In specific, some database systems which are popular that support keyword search are BANKS, DBXplorer, SPARK and discover. These systems concentrate on generating sub graph of tuples, where smaller the sub graph, better it is ranked. However these approaches do not possess fine grained control over user intent that parsing with a grammar allows. And also, to the best of our knowledge, none of these methods allow keywords such as, from, to, etc. or noise tokens in the query.

#### *4) Web search*

There has been some work done on Query segmentation [23, 24] using query logs and click logs. Nevertheless, these do not exploit the rich structured information in the database, and also do not consider relational constraints.

#### *5) Query expansion and Approximate match*

Recent work [25] has also studied at expanding the concept between a section of the search query and a concept in underlying information. Other works which handle the issue use query cleaning, query substitutions rewrite rules and approximate entity extraction. These works are statistically independent to the work and can be used to specifically recognize user goal even better.

#### *6) Query classification and understanding user goal*

Recent work has been done on query goal classification. [26, 27, 28**]** The aim of the work is to recognize whether a search query belongs to a specific stream or not. The output of a classification system is based on absolute value which can be used, for eg: to direct the query to an approximate search vertical. Along with this, the outputs of a query extracting system is more detailed one and the kinds of problems and challenges are examined by the two issues which are basically different. It may be noted that query classification and query parsing can complement one another since query parsing is useful and relevant even after we slow down the search vertical using query classification

*7) XPATH Streaming*

There has been much work done on recognizing the matches from a given set XPaths [29,30]. Most previous work in this area builds first automata and then streams the XML information via the automata. Nevertheless, this work concentrates on exactly matching the given XPath, which does not have any underlying information or relational constraints, and thus requires various techniques.

*8) Frequent Itemsets*

Eventually, we note that the relational matching pattern issue is related to the well known problem of finding frequent item sets [31, 32]. While the work is same, information specific to the problem is used to carefully choose hitting sets and obtain an algorithm that under practicable presumption is worst case optimal. A toolkit called as "Learning Pinocchio (*LP*)2",[1] was applied on CVs to extract and learn the rules from the CVs.. The document recognized in their work includes a structure of City, Name, Province, Email Telephone, Fax and Zip. Learning Pinocchio is an adaptable technique for Information extraction, which is based on transformation like rule learning. Rules are learned by postulating over examples marked via XML tags in training set.

A proposal has been propound in [2] for CV text parsing to support automatic CV routing and management. A stream of documents is extracted and framework is designed. In the first phase, a resume is dismembered into series of tagged blocks which indicate the information types. In the second phase, the detailed document, such as Address, Name etc. are recognized in blocks instead of searching on whole CVs. Based on the necessities of recruitment team that integrates the database with Information Extraction technologies and resume routing, general information fields like Education, Personnel information and others are defined .

Resume RDF ontology was introduced to model resume using RDF model. Work also extended to FOAF [4] which supports description of resume such as publication property to describe information about publications. ResumeRDF has rich set of classes and properties to describe resume information. ResumeRDF [5] expresses resume information using two namespaces

▪http://captsolo.net/semweb/resume/0.2/cv.rdf - Resume ontology

▪http://captsolo.net/semweb/resume/0.2/base.rdf - Property value taxonomy

Using ResumeRDF, a rich set of resume information can be described such as a person's detailed information, his/her skill information, reference information, education and work experiences etc. Authors also examined about finding and aggregating the resume information on the web using online and social networking sites. However, using community and social networking sites, web suffers from information reusing and sharing. Also they contend that data fusing from various sources will remain a problem. People can send resumes from their shared or personal accounts and referenced candidates. Also the applications cannot depend on such sites to display and consume resume document as Linked Data allows SPARQL query service to access the resources and connects distributed data sources across the web. Linked Data discovers and integrates resume information which brings high benefits to the organizations and people too.

In [7] an ontology hybrid approach was presented which effectively matches job seekers skills by using the traceable model to identify the kind of match required by an employer.

The authors in [8] proposed a technique for retrieving information of resume via Linked Data which enabled the web to share data among various data sources and to find any kind of document. In [9], a technique was developed for automated resume document extraction to support rapid resume management and search which extracts several important informative fields using natural language processing(NLP). The authors in [10] found out a systematic solution of text retrieval in online chinese CVs using statistical algorithms and rule base to extract information. In [11], the authors researched on resume block analysis based on identifying multilevel information, matching patterns and developed the biggest resume parser system.A technique was developed in [12] to convert the resume into ontological structural model, which gives an efficient way of searching resumes in English and Turkish. In [13] work was proposed using ontology for information extraction from resumes using the WordNet for calculations.

The authors in [14] proposed a technique which represents Web information via WordNet and vector model and done text summarization and personalization for data. In [15] two approaches were developed to extract information like Dependency Based Approach and Rule Based Approach on the collection of recipe documents. This kind of semantic annotation is very useful for efficient answering of search queries, text summarization, clustering and so on. Work was proposed in [16] based on query web search system by using the personal document of users where users can get relevant web pages based on their selection from the area of interest. The literature survey also says that existing websites offer more options for search like keyword search, domain search, location etc for document retrieval. But they have some demerits too; For ex, LinkedIn [17] retrieves pages which are mostly irrelevant. Resume builder [18] and Indeed.com provide very minimum number of resumes. For example, when the keyword 'C' is used for search, all the resumes that have 'C' as the initial are also listed; whereas the actual requirement is the programming language 'C'.

## III. SYSTEM ARCHITECTURE

System architecture describes the behaviour and structure of a system. It explains system components and provides a plan to implement the work.

### A. *Problem Statement*

Today recruiters find their suitable human resources by searching job related web sites. In the same way, the job seekers also select their suitable job by posting their resumes across multiple web sites. The work attempts "To produce a listing of resumes along with the ranks, for a given job description". The ranks play the role of indicators and aid the decision maker in the recruitment process.

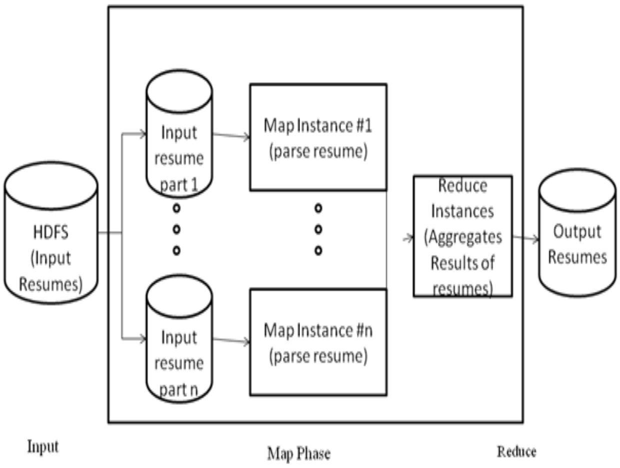
### B. *Objectives*

The project addresses the problem of finding candidate resume(s) that match a job requirement. The work accomplishes the following objectives:

- Collection of resumes from various websites.
- Preprocessing of Resumes and converting into text files.
- Running text files in Hadoop, and calculating the overall coverage of Resumes.
- Summing up the Comprehensibility.
- Ranking of Resumes according to Coverage and Comprehensibility.
- Finally, given a job description listing of all the resumes along with ranks.

The system architecture is depicted in Fig.1

*Fig 1. System Architecture*


HDFS (Input Resumes)
Input resume part 1
Input resume part n
Map Instance #1 (parse resume)
Map Instance #n (parse resume)
Reduce Instances (Aggregates Results of resumes)
Output Resumes
Input
Map Phase
Reduce

The System architecture has three modules.

- Resume Parser
- Concept Builder
- Analysis

There are two phases in system architecture.

- Training Phase
- Test Phase

#### 1) *Training Phase*

In training phase reference resumes which are properly written with all the fields are given as input to the Resume Parser. From each resume, fields are parsed to build a concept map. After building a concept map the concept map is fed into the database.

#### 2) *Test Phase*

In test phase, test resumes are given as input to the parser. The resume is parsed and concepts are extracted. Concepts are mapped to those in the training concepts to find out whether all the concepts are covered or something is missing.

### C. *Module Level Description*

#### 1) *Resume Parser*

In resume parser, the input given to the parser should be in pdf format, because parsing is difficult with MS-word. The pdf file is converted into the text file followed by extraction of nouns.

A check is made to see whether noun is standing alone or co-occurring with other noun in a sentence. If noun is standing separately then it is taken as Header/Title or if noun is co-occurring with some other noun in the line then it is a key value pair.

*2) Concept Builder*

Once the body section of resume is found, we have to build the concept map. For building concept map we have to give very good resumes, resumes which are fully covered.

*3) Analysis*

Analysis module matches to concept tree `a section in resume and finds if any missing elements are there in section and give results.

- *Computing Coverage*

The work is a qualitative approach where we compute for each resume, its coverage level. For this, information from the resumes is extracted and concepts and sub concepts are identified. Coverage checks to see whether a resume is complete or fully covered.

- *Computing Comprehensibility*

This approach is based on past work. It determines whether learning concepts described in resumes are presented in a coherent and cohesive manner. It requires parsing of the text in all sections of the resumes and identifying the concepts of each section and recognizing which concepts are related to each other and in what manner.

## IV. IMPLEMENTATION DETAILS

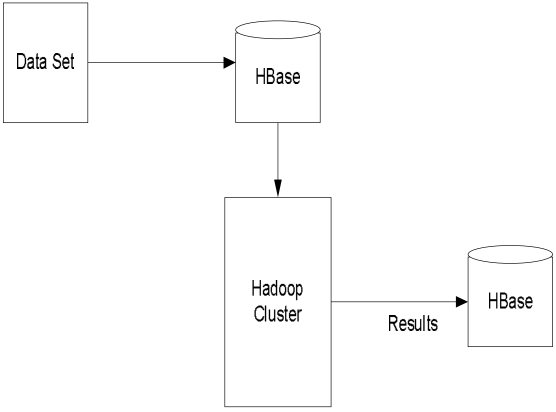
### *A. Data collection*

Resumes are collected from various websites like naukri.com, LinkedIn etc. These websites have CV's which are uploaded by the applicant to seek a job. The uploaded CV's are in different formats like DOC and PDF format. Retrieving and analyzing data from PDF files is easier when compared to DOC files, as word parsing is difficult and there are no free API's present. So, DOC files are converted into PDF files first, and then pdf file is converted to text file using Resume parser. Nouns are then extracted from the text file document by using Maxent Tagger.

### *B. Running Map Reduce Jobs*

After conversion and extraction of resumes, we have to build the concept map which is done using the Hadoop framework based on MapReduce model.

*Figure 2 describes the hadoop implementation view.*



### *C. Hadoop Map Reduce*

Apache Hadoop is a software platform based on Map Reduce model for running of large number of document sets in a distributed fashion on enumerate clusters of commodity hardware. The platform takes care of monitoring and scheduling tasks and re-implements any failed tasks. Map Reduce is a platform for running parallelizable problems across a large number of nodes, which may be a cluster or a grid. Processing occurs on data stored either as structured data set or unstructured dataset. Map Reduce can take advantage of locality of data, running it near or on the storage benefits in order to decrease the distance over which it must be transferred.

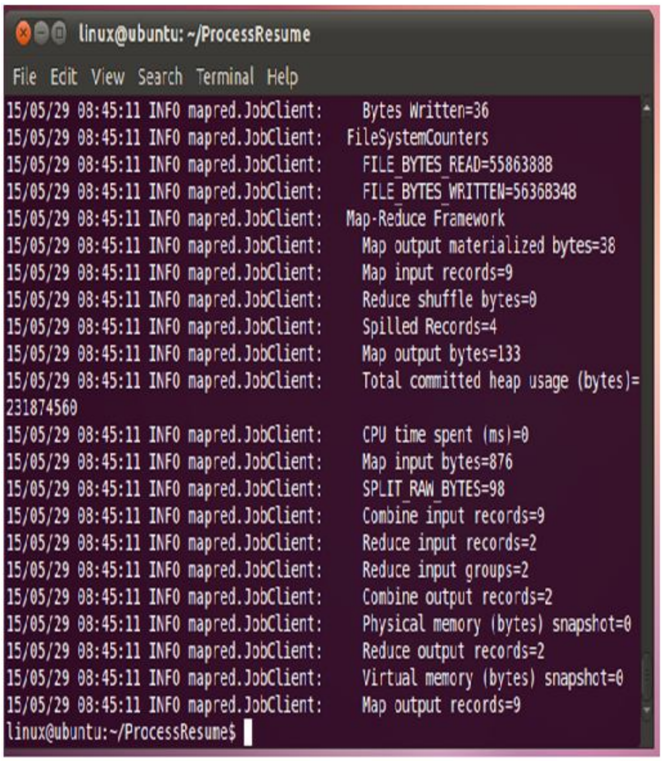
1) *Map step:* Each user node pertains to the map( ) function that works on the local data and writes the results temporarily to the storage. A master node adapts that for dispensable copies of input data, only one time it is run.

2) *Shuffle step:* User nodes redistribute the data based on the keys generated by the map() function, such that all data belonging to one key is located on the same user node.

3) *Reduce step:* User nodes now run each group of output data, in parallel for each key.

## V. EXPERIMENTAL RESULTS

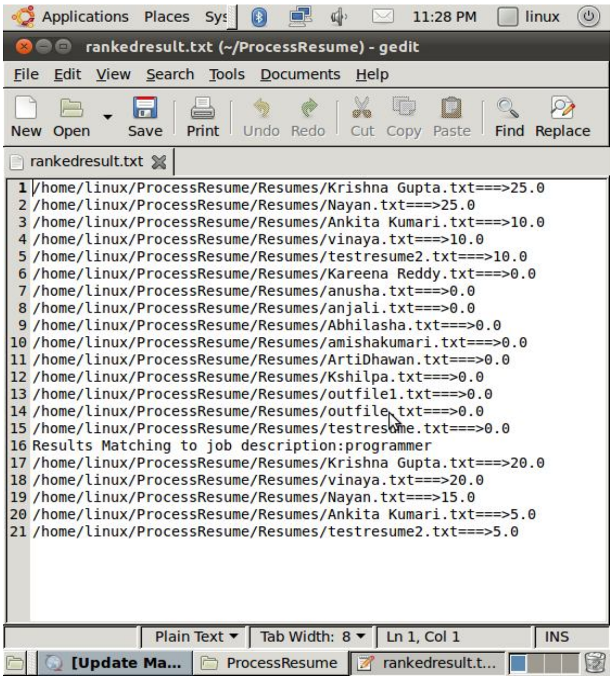
- In this section, the results of the work are presented. To evaluate the performance, the work was applied on real data set of resumes, containing around 200 resumes of postgraduate and undergraduate students. We present our experiment based on resume information where a dataset containing description of resume information is generated and then search queries are used to find the required information from the resume. This dataset is assumed as a part of the Web of Data and is used for our experimental purpose. A CV dataset contains information about business and academic information. Similarly, it shows other information such as skill, company, title etc.
- Resume processing like building concepts and analyzing phase is done in Hadoop as shown in fig. 3. To build concepts of resumes, multiple resumes are processed through map reduce jobs. The qualitative measure 'coverage' can be taken as value denoting how many sections and sub sections are covered in a resume. The next job was then to see how many concepts from a class are covered in a resume. This required extraction of concepts and performing matching. If all the concepts are covered in the resume, as per resume grammar, then that particular resume is considered as complete resume or else incomplete. To compute comprehensibility, each line in resume is fed to mapper which splits each line with (.) as separator and gives comprehensibility of that line. Reduce sums up all the results and gives comprehensibility of each resume as output.


linux@ubuntu: ~/ProcessResume
File Edit View Search Terminal Help

```
15/05/29 08:45:11 INFO mapred.JobClient:     Bytes Written=36
15/05/29 08:45:11 INFO mapred.JobClient:   FileSystemCounters
15/05/29 08:45:11 INFO mapred.JobClient:     FILE_BYTES_READ=55863888
15/05/29 08:45:11 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=56368348
15/05/29 08:45:11 INFO mapred.JobClient:   Map-Reduce Framework
15/05/29 08:45:11 INFO mapred.JobClient:     Map output materialized bytes=38
15/05/29 08:45:11 INFO mapred.JobClient:     Map input records=9
15/05/29 08:45:11 INFO mapred.JobClient:     Reduce shuffle bytes=0
15/05/29 08:45:11 INFO mapred.JobClient:     Spilled Records=4
15/05/29 08:45:11 INFO mapred.JobClient:     Map output bytes=133
15/05/29 08:45:11 INFO mapred.JobClient:     Total committed heap usage (bytes)=
231874560
15/05/29 08:45:11 INFO mapred.JobClient:     CPU time spent (ms)=0
15/05/29 08:45:11 INFO mapred.JobClient:     Map input bytes=876
15/05/29 08:45:11 INFO mapred.JobClient:     SPLIT_RAW_BYTES=98
15/05/29 08:45:11 INFO mapred.JobClient:     Combine input records=9
15/05/29 08:45:11 INFO mapred.JobClient:     Reduce input records=2
15/05/29 08:45:11 INFO mapred.JobClient:     Reduce input groups=2
15/05/29 08:45:11 INFO mapred.JobClient:     Combine output records=2
15/05/29 08:45:11 INFO mapred.JobClient:     Physical memory (bytes) snapshot=0
15/05/29 08:45:11 INFO mapred.JobClient:     Reduce output records=2
15/05/29 08:45:11 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=0
15/05/29 08:45:11 INFO mapred.JobClient:     Map output records=9
linux@ubuntu:~/ProcessResume$
```

*Fig. 3 Map Reduce jobs Processing each Resumes*

- Once Map Reduce jobs are done it gives summary of each resume called as Comprehensive Quality. It gives summary on missing information of header and data. After summing up the comprehensibility, each resume is ranked according to the coverage and comprehensibility. Finally, for a given job description it lists all the resumes along with the ranks. Figure 4 is a snapshot showing output of ranked resumes.


Applications Places Sys
11:28 PM
linux
rankedresult.txt (~/ProcessResume) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
rankedresult.txt
Plain Text
Tab Width: 8
Ln 1, Col 1
INS
[Update Ma...
ProcessResume
rankedresult.t...

```
/home/linux/ProcessResume/Resumes/Krishna Gupta.txt===>25.0
/home/linux/ProcessResume/Resumes/Nayan.txt===>25.0
/home/linux/ProcessResume/Resumes/Ankita Kumari.txt===>10.0
/home/linux/ProcessResume/Resumes/vinaya.txt===>10.0
/home/linux/ProcessResume/Resumes/testresume2.txt===>10.0
/home/linux/ProcessResume/Resumes/Kareena Reddy.txt===>0.0
/home/linux/ProcessResume/Resumes/anusha.txt===>0.0
/home/linux/ProcessResume/Resumes/anjali.txt===>0.0
/home/linux/ProcessResume/Resumes/Abhilasha.txt===>0.0
/home/linux/ProcessResume/Resumes/amishakumari.txt===>0.0
/home/linux/ProcessResume/Resumes/ArtiDhawan.txt===>0.0
/home/linux/ProcessResume/Resumes/Kshilpa.txt===>0.0
/home/linux/ProcessResume/Resumes/outfile1.txt===>0.0
/home/linux/ProcessResume/Resumes/outfile.txt===>0.0
/home/linux/ProcessResume/Resumes/testresume.txt===>0.0
Results Matching to job description:programmer
/home/linux/ProcessResume/Resumes/Krishna Gupta.txt===>20.0
/home/linux/ProcessResume/Resumes/vinaya.txt===>20.0
/home/linux/ProcessResume/Resumes/Nayan.txt===>15.0
/home/linux/ProcessResume/Resumes/Ankita Kumari.txt===>5.0
/home/linux/ProcessResume/Resumes/testresume2.txt===>5.0
```

*Fig 4 Output of Ranked resumes*

## VI. CONCLUSION

In this work, a qualitative assessment of resumes on the basis of different quality parameters using a simple text analytic based approach for a resume collection was described. The resume collection is assessed for two qualitative aspects, coverage and comprehensibility; and these ratings are transformed into a comprehensive quality rating. All the three parameters are collectively measured into a combined 1 to 5 rating scale for associating a quality metric for resumes. The qualitative evaluation results obtained through the algorithmic approach are congruent to and are hence validated through the wisdom of crowds. Although we evaluated and combined two qualitative parameters for resume assessment in a systematic and thorough manner, some improvements and extensions may still be possible. The pdf parsing and section identification can also be improved. The standard reference documents can be augmented as well. Transforming the computed values to ranks has been the trickiest part. While for coverage, it is simpler; but in case of comprehensibility, it is a bit complex and tricky to transform computed values to 1 to 5 scale rating. Nevertheless, the algorithmic formulation has the possibility of being used in an annotation and recommendation system.

## VII. FUTURE WORK

In the future, work may be done to make use of resume information to better the recruitment process and decrease the costs of making a new hire. A lot of the information that is needed is contained in those accurately or not so accurately crafted resumes and the good news for all, is that, there is the technology to parse the resumes. So, first and foremost step is that all the relevant factual information from the collected resumes be taken off and input those invaluable factual information into the database. Second, recruiters should routinely trace the source of each seeker in order to measure the success of their hiring effort. There is also an abundance of ‘brilliance’ that can be gathered about qualification trends and skill sets. To be much competitive, seekers must use their resumes to advertise their strengths, and also highlight their academic and professional education. If this data is consistently normalized, parsed and stored, the movement of competencies in the business can be seen, and their respective demand trends and supply. Such data, when properly combined, and organized would create a pool of knowledge for the big project or business line.

## REFERENCES

[1] Ciravegna, F., Lavelli, A.: Learning pinocchio: adaptive information extraction for real world applications. Nat. Lang. Eng. 10(2), 145–165 (2004)

[2] Yu, K., Guan, G., Zhou, M.: Resume information extraction with cascaded hybrid model. In: ACL 2005: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 499–506. Association for Computational Linguistics(2005)

[3] Uldis Bojars, John G. Breslin, *"ResumeRDF: Expressing Skill Information on the Semantic Web"*. The 1st International ExpertFinder Workshop (EFW 2007), Berlin, Germany, 16 January 2007 .

[4] Uldis Bojars, *“Extending FOAF with Resume Information”*.Available http://www.w3.org/2001/sw/Europe/events/foafgalway/papers/pp/extending_foaf_with_resume/

[5] ResumeRDF, available at http://rdfs.org/resume-rdf/

[6] FOAF ontology, avialable at http://xmlns.com/foaf/spec/

[7] Maryam Fazel-Zarandi1, Mark S. Fox2, “Semantic Matchmaking for Job Recruitment: An Ontology-Based Hybrid Approach”, International Journal of Computer Applications (IJCA), 2013.

[8] Ujjal Marjit, Kumar Sharma and Utpal Biswas, “Discovering Resume Information Using Linked Data”, in International Journal of Web & Semantic Technology, Vol.3, No.2, April 2012.

[9] Kopparapu S.K, “Automatic Extraction of Usable Information from Unstructured Resumes to aid search”, IEEE International Conference on Progress in Informatics and Computing (PIC), Dec 2010.

[10] 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.

[11] 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.

[12] 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.

[13] Di Wu, Lanlan Wu, Tieli Sun, Yingjie Jiang, “Ontology based information extraction technology”, International Conference on Internet Technology and Applications (iTAP), Aug 2011.

[14] Salah T. Babekr, Khaled M. Fouad, Naveed Arshad , “Personalized Semantic Retrieval and Summarization of Web Based Documents”, in International Journal of Advanced Computer Science and Applications Vol. 4, No.1, 2013.

[15] Zeeshan Ahmed, “Domain Specific Information Extraction for Semantic Annotation”, A Master thesis from Joint European Program, 2009.

[16] Mahendra Thakur, Yogendra Kumar Jain, Geetika Silakari “Query based Personalization in Semantic Web Mining”, International Journal of Advanced Computer Science and Applications, Vol. 2, No.2, February 2011.

[17] https://www.linkedin.com

[18] www.resume.com

[19] www.indeed.com

[20] R. Fagin, B. Kimelfeld, Y. Li, et al. Understanding queries in a search database system. In PODS, pages 273–284, 2010.

[21] A. Chandel, P. C. Nagesh, and S. Sarawagi. Efficient batch top-k search for dictionary-based entity recognition. In ICDE, page 28, 2006.

[22] Y. Chen, W. Wang, Z. Liu, et al. Keyword search on structured and semi-structured data. In SIGMOD Conference, pages 1005–1010, 2009.

[23] N. Mishra, R. S. Roy, N. Ganguly, et al. Unsupervised query segmentation using only query logs. In WWW, pages 1–92, 2011.

[24] M. Hagen, M. Potthast, B. Stein, et al. Query segmentation revisited. In WWW, pages 97–106, 2011.

[25] T. Cheng, H. W. Lauw, and S. Paparizos. Fuzzy matching of web queries to structured data. In ICDE, pages 713–716, 2010.

[26] X. Li, Y.-Y. Wang, D. Shen, et al. Learning with click graph for query intent classification. ACM Trans. Inf. Syst., 28(3), 2010.

[27] J. Hu, G. Wang, F. H. Lochovsky, J.-T. Sun, and Z. Chen. Understanding user’s query intent with wikipedia. In WWW, pages 471–480, 2009.

[28] A. Ashkan, C. L. A. Clarke, E. Agichtein, et al. Classifying and characterizing query intent. In ECIR, pages 578–586, 2009.

[29] T. J. Green, A. Gupta, G. Miklau, et al. Processing xml streams with deterministic automata and stream indexes. ACM Trans. Database Syst., 29(4):752–788, 2004.

[30] Y. Diao, P. M. Fischer, M. J. Franklin, et al. YFilter: Efficient and scalable filtering of xml documents. In ICDE, pages 341–, 2002.

[31] D. Gunopulos, R. Khardon, H. Mannila, et al. Discovering all most specific sentences. TODS, 28(2), 2003.

[32] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In SIGMOD Conference, pages 207–216, 1993.

[33] ReadabilityScores,http://www.trans4mind.com/personal_development/writing/Readability_software/flesch.htm.

[34] Vinaya Ramesh Kudatarkar, Manjula Ramannavar, Dr. Nandini S.Sidnal: "A Survey on Unstructured Text Analytics Approaches for Qualitative Evaluation of Resumes" in International Journal of Emerging Technology in Computer Science and Electronics (IJETCSE) April 2015.

[35] Sumit Maheshwari, Abhishek Sainani, and P. Krishna Reddy "An Approach to Extract Special Skills to Improve the Performance of Resume Selection" Center for Data Engineering, International Institute of Information Technology, Hyderabad, (IIIT-H).