International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 14 Issue 2 –APRIL 2015.

A Survey on Unstructured Text Analytics Approaches for Qualitative Evaluation of Resumes

VINAYA RAMESH KUDATARKAR[#1], MANJULA RAMANNAVAR[*2], DR. NANDINI S.SIDNAL[*3]

[#1]*M.Tech, Dept. of CSE, KLS Gogte Institute of Technology, Belagavi, Karnataka, India*

[*2]*M.Tech, Dept. of CSE, KLS Gogte Institute of Technology, Belagavi, Karnataka, India*

[*3] *Ph.D., Dept. of CSE, KLE's, College of Engineering and Technology Belagavi, Karnataka, India*

1vinaya0210@gmail.com

2manjular@git.edu

3sidnal.nandini@gmail.com

***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, the ability to summarize, analyze which make sense of such data for making good business decisions This paper reviews how to organize and understand the textual data and presents an unstructured text analytics approach for qualitative evaluation of CV/Resume documents. This paper proposes an effective approach for extracting the resume information from websites and analyzing it thereby making the job easier for finding suitable resumes. The survey results in obtaining fair measure of qualitative account of a resume document on the parameters of coverage, readability and comprehensibility 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 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, the ability to summarize, analyze which make sense of such data for making good business decisions. For companies and industries it is very challenging to understand and analyze the 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 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 human and computer language. The field of natural language processing has come up with many technologies which teach computers natural language where computers understand, analyze and come up with new text. The technologies which have been developed are information extraction, categorization, concept linkage, summarization, clustering, topic tracking, information visualization, question answering and association analysis.

Data mining is a related technology whose tools are designed to handle only structured data from databases. On other hand , text analytics can handle both semi-structured and unstructured data sets. Humans have the ability to understand and distinguish natural language and apply linguistic patterns to documents and to overcome the obstacles which computers cannot handle such as contextual meaning, spell checking etc.

Starting with the collection of texts, a text analytic tool will retrieve particular text and preprocess it by character set and format. Then it is sent to the analyze phase, where information is extracted repeatedly. Analytic techniques, such as clustering, summarization, etc. could be used depending on the needs of the organization.

The rest of the paper is organized as follows: Section II provides the background and discusses various types of resume parsers. Section III presents our views and perceptions as observations. Section IV describes related work. Section V illustrates the basic idea of the proposed approach. Section VI summarizes the paper in the form of conclusion.

II. BACKGROUND

A career for graduates in human resource is all about getting placed in a good organization where best people work and

where their rights are protected and benefits are looked after. Recruiters will be looking for such individuals who look after the organization's functioning like pay, benefits, safety etc. Finding a good job in HR is not that easy, there are many graduate schemes which take large number of graduates into their organizations. 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 crack interview. There are many types of resumes like chronological resumes, visual resumes, functional resumes etc.

Statistics indicate that an employer hardly spends a lot of time in looking at the resumes, so it should be impressive and imperative. In this competitive world, a candidate should possess a powerful resume which conveys the required information in a manner that it stands out among resumes of contemporaries. . On the database of companies there will be lakhs together resumes which will be in unstructured and free style resumes in MS word document. The information and the structure contents of resumes will be collection under sub topics, the classification and the representation of information will be totally different in all resumes. So ,gathering the data from each resumes and storing it into the companies database in a particular format will reduce some human effort. There are some difficulties of resume service by the unions or commercial companies which consume too much of time, capacity, money, human effort and so on. These companies require filtered/parsed resumes for employment in HR department of the commercial companies.

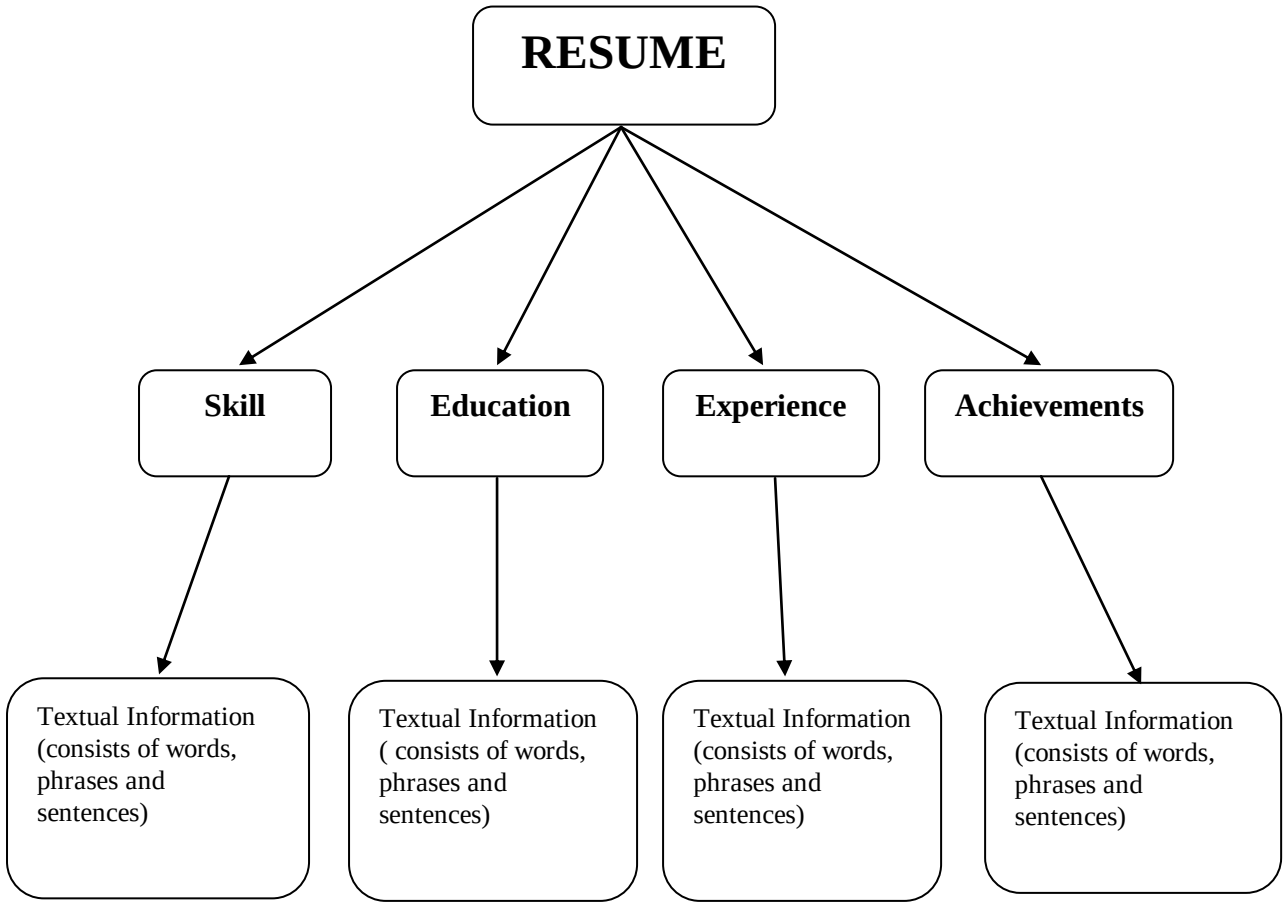

Fig.1. Hierarchical structure of Resume

Fig 1 presents Structure of Resume. Each resume contains different sections like skills, education, etc and each section contains of words and sentences as features. The top layer which is depicted as "Layer 0" is resume . It can be observed that it has sections like experience, education, achievements and skills form the first layer "Layer 1" of the resume. Each section is presented by the text containing words and phrases in sentences which gives the second layer "Layer 2" of the resume. Based on the structure of the text, the text of each section in the second layer is organized into various layers.

Job seekers post their resumes on various websites like sdIndeed.com, Linkedin, Naukri.com, Monster.com, Resume builder etc. Certain websites retrieve unwanted documents from the resumes, say like Linkedin[17] retrieves the which are mostly irrelevant. Resume builder[18] website provides very minimum number of resumes and etc. For example When the keyword 'C' is used for search all the resumes whose has 'C' as the initial is also listed; but whereas the actual requirement is the programming language 'C'. So here comes a picture of Resume Parsing.

RESUME PARSING:

Resume parsing is the process of analyzing the document and extracting the elements or concepts of the resume, of what the writer actually meant to say like his skills, education, experience and 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.

There are some different types of parsers. Basically there are three types of approach of parsing resumes.

- Keyword Parsers
- Grammar Parsers
- Statistical Parsers

KEYWORD PARSERS:

Keyword parsers are the a parsers which works by identifying phrases and words and very simple patterns in the documents and then applies simple algorithms to documents. These give least accuracy which has 70% accuracy rate, and its very hard to get beyond 70%, because these type of parsers cannot extract the information which doesn't surrounds the keywords
and there keyword which are ambiguous in nature. These type of parsers make wrong interpretation too.

GRAMMAR PARSERS:

It is the process of parallelizing the sequence of words with it formed grammar. The result will be in syntactical tree or parse tree. Grammar parsing use Context free Grammar and these are much more difficult to extract and analyze and these are not enough to expressive. These parsers capture every meaning of the sentence in the resume. These are bit complicated parsers than keyword parsers, because they capture much more detail of the document and they are also able to differentiate between the different meaning that one phrase and word which may have different contexts. Its accuracy level is above 90%,and it requires a lot of manual encoding by lot of testing and skilled language engineer and make sure that do not degrade the performance.

STATISTICAL PARSERS:

Statistical Parsers lie in between Grammar parsers and keyword parsers. This type of parser make an effort to apply on numerical models of documents to identify structure of resumes. It is same like grammar parser they can differentiate between different context of same phrase or word and also captures variety of structure such as timeline, address, achievements etc. It actually lies between keyword parser and grammar parser. Statistical parser needs trained data to be accurate which is expected to process.

III. RELATED WORK

Related work can be placed into 8 categories.

Parsing-based Keyword Search: Fagin [20] introduced the problem of keyword search based on parsing; that is, given a grammar, a database of documents and queries, the goal was to return the most clearly defined parse. It was shown that this issue was hard to deal with; however, they found that the issue of returning all apropos parses has complexity in the input and result. Subsequently, they rewrote rules, and showed that under some conditions, finding all rewritten parses is decidable. The proposed work focuses on efficiency, i.e., how to design algorithms and index structures to efficiently support a general parsing-based keyword search. The proposed work differs from prior work in the use of a scoring function to express parse preferences. There are two advantages to the scoring function approach compared to the containment approach: (1) Containment based ordering is shown to be NP-hard, while scoring function ordering is manageable for many interesting classes of functions that are considered. (2) Scoring functions impose a total ordering over the parses unlike containment, e.g., the top-k parses are explicitly defined.

Dictionary-based Segmentation: Chandel [21] considered the issue of string segmentation while sharing search for the structured document. However, earlier work did not parse with grammar. The proposed work uses their algorithms to improve the efficiency of containment lookups. Another recent work focuses on query segmentation using CRFs. They use a notion similar to maximal matches to establish matches that also appear in the database. However, note that this work also does not consider relational constraints, and does not support a grammar. There has also been some work on annotating web queries. Once again, this work does not consider relational constraints or a grammar, and restricts matches to be tokens from a single table. However, note that some of their probabilistic techniques could be utilized to infer patterns.

General Keyword Search: There has been a large body of work on Keyword Search [22] in databases over the last decade. In particular, some popular database systems that support keyword search include DBXplorer, SPARK, BANKS and Discover. These systems focus on generating a connected sub graph of tuples, where the smaller the subgraph, the better it is ranked. However, these approaches do not possess fine-grained control over user intent that parsing with a grammar allows. Also, to the best of our knowledge, none of these approaches allow special keywords (e.g., from, to) or noise tokens in the query.

Web Search: There has been some work recently on query segmentation [23, 24] by using click logs and query logs. However, these approaches do not exploit the rich structured information in the database, and do not consider relational constraints.

Query Expansion and Approximate Match: Recent work [25] has also looked at expanding on the notion of a "match" between a portion of the search query and a concept in underlying data. Other papers tackle the problem using rewrite rules, query substitutions, query cleaning, and efficient approximate entity extraction. These papers are orthogonal to the proposed work (which focuses on efficiency of the basic parsing infrastructure) and can be used to potentially identify user intent even better.

Query Classification and Understanding User Intent: There has been a lot of recent work on query intent classification. [26, 27, 28**]** The goal of this work is to identify if a search query belongs to a particular category or not (e.g., navigational v/s informational, travel v/s no travel). The output of an intent classification system is a categorical value that can be used, for example to direct the query to an appropriate search vertical. In contrast, the output of a query parsing system is more detailed and the kinds of challenges and issues studied by the two problems are fundamentally different. It may be noted that query classification and query parsing can complement one another since query parsing is relevant and useful even after we narrow down the search vertical using query classification.

XPath Streaming: There has been some work on identifying matches from a given set of XPaths[29,30]. Most prior work in this area builds automata and then "streams" the XML data through the automata. However, this body of work focuses on exactly matching the given XPath, does not have any underlying data or relational constraints, and therefore requires different techniques. Similar work has addressed the problem of retrieving and indexing regular expressions. Once again, this work does not address the problem of matching against an auxiliary database in addition to a grammar.

Frequent Itemsets: Finally, we note that the relational matching problem is related to the well-known problem of finding frequent item sets [31,32]. While the proposed work is similar, information specific to the problem is used to carefully choose hitting sets and obtain an algorithm that under reasonable assumptions is worst case optimal.

A toolkit named as "Learning Pinocchio (*LP*)2",[1] was applied on resumes to learn and extract the rules from resumes. The document recognized in their work includes a structure of City, Name, Email, Province, Telephone, Zip and Fax code. 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 through XML tags in a training body.

An approach has been propound in [2] for resume information extraction to support automatic resume routing and management. A stream of documents is extracted and framework is designed. In the first phase, a resume is dismembered into series of blocks attached with tag indicating the information types. In the second phase, the detailed document, such as Address, Name etc. are recognized in certain blocks, instead of searching in the whole resume. Based on the necessities of recruitment management team which integrates the database construction with Information Extraction technologies and resume routing, general information fields like Education, Personal information and others are defined.

Uldis Bojars introduced ResumeRDF ontology 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.

Maryam Fazel-Zarandi [7] also proposed an ontology hybrid approach which effectively matches job seekers skills by using the traceable model to identify the kind of match required by an employer. Ujjal Marjit [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. Koparappu of TCS Innovations lab [9], developed a technique for automated resume document extraction to support rapid resume management and search which extracts several important informative fields using natural language processing(NLP). Zhi Xiang Jing [10] found out a systematic solution of the document retrieval in online Chinese resume using statistical algorithms and rule base to extract information. Zhang Chuang [11], researched on resume block analysis based on identifying multilevel information, matching pattern and developed the biggest resume parser system.

Celik [12], developed a technique to convert the resume into ontological structural model, which gives an efficient way of searching resumes in English and Turkish. Di Wu [13] proposed work using ontology for the information extraction from resumes using the WordNet for calculation. Salah T. Babekr [14] proposed a technique which represents Web information via WordNet and vector model and done text summarization and personalization for data.
Zeeshan Ahmed [15] developed two approaches to extract information like Dependency Based Approach and Rule Based Approach on the collection of recipe documents. This kind of semantic notation is very useful for efficient answering of search queries, text summarization, clustering and so on. Mahendra Thakur [16], proposed a work 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 like - 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; but whereas the actual requirement is the programming language 'C'.

IV. OBSERVATIONS

The problem here is to develop an approach to select the proper resumes efficiently. It is observed that there are common attributes which are present in all the resumes in the group and also each resume may obtain some special features that could differentiate it from all of the other resumes in the group. The instinct here is that if the special attributes of each resume are recognized, the time required for selecting an proper resume can be reduced in comparison to the time required by taking into account all the information. Normally, each resume is presented by a document and the information in the resume is splitted into different categories. Special information of a resume entails special information in each section. For example, there could be specialty in specialty in achievements, skills, specialty in education etc. The issue here is to recognize the special information from each resume. Each section contains different types of information. For example experience section contains long sentences,) skill values (C++, java) and skills section contains skill type (programming skills). Thus the development point of view to process the resume data is a cumbersome, as separate approaches have to be developed by recognizing special information for each type of information and merge the same properly. The main problem here is how to compute the specialness of text for all the resumes. The work extends to the concept of special attributes to propose an productive approach for the resume selection issue.

V. PROPOSED WORK

The proposed model says – (i) Statistical mapping for categorizing the resumes and (ii) Ranking of resumes. This model of storing the resume information retrieves the relevant resumes efficiently.

- The main goal of this work is extracting the resume information which makes the job easier by finding the suitable resume to fit their needs.

- The two key measurements that we look for, in resume parsing are Accuracy and Coverage level. Coverage level reports what a parser actually tries to extract and Accuracy reports how well a parser can identify information from a Resume**.**

The proposed system architecture has three modules.

- Resume Parser
- Concept Builder
- Analysis

There are two phases in system architecture.

- Training Phase
- Test Phase

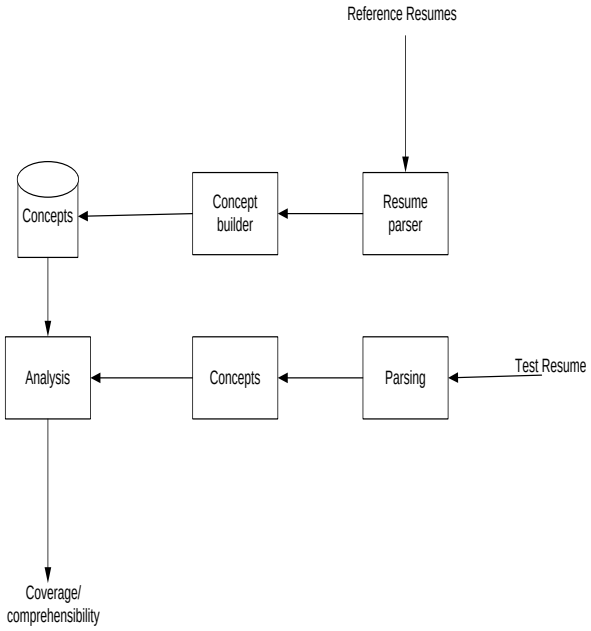

Fig 2: Proposed System

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

Test Phase: In test phase, test resumes are given as input to the parser; it will parse the resume and extract the concepts from resume. Concepts are mapped over here and then found out whether the concepts are covered or something is missing out and then analysis is made.

MODULE LEVEL DESCRIPTION

Resume Parser: In resume parser, the input given to the parser should be in pdf format, because word parsing is difficult. First, pdf file is written into the text file and nouns are extracted. Then a check will be made to find out whether noun is standing alone or co-occurring with the 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.

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.

Analysis: Analysis module will match to concept tree a section in resume and find if any missing elements are there in section and give result.
In analysis we focus on three more concepts that is Readability, Coverage and Comprehensibility

Computing Readability: This approach based on earlier work [33] tries to capture the syntactic complexity of the text written in resume and aims at word and sentence structure.

Computing Coverage: It is a qualitative approach where we compute each resume for its coverage level. For this we need to extract information from the resumes and identify the concepts and sub concepts

Computing Comprehensibility: This approach is based on past work. It amounts 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.

VI. CONCLUSION

The work contributes to the discussion presenting a resume parser in which the grammar and probabilistic parameters are induced from a tree bank and have shown that its performance is superior to previous parsers in this area. An experiment that suggests that its superiority stems mainly from unsupervised learning plus the more extensive collection of statistics that it uses, both more and less detailed than those in previous systems. The proposed model collects resumes through web search and ranks based on cosine similarity measure. Statistical parsing plays vital role while extracting and keeping the information relevant and up-to-date. Thus the search time for required document is reduced when data is stored. The model also reduces the human effort required in seeking the relevant information.

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 at 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 91–92, 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.