Ontology based ranking of documents using Graph Databases: a Big Data Approach

A.M. Abirami
Dept. of Information Technology
Thiagarajar College of Engineering
Madurai, Tamil Nadu, India
abiramiam@tce.edu

A. Askarunisa
Department of CSE
Vickram College of Engineering
Enathi, Tamil Nadu
nishanazer@yahoo.com

R.Sangeetha, C.Padmavathi, M.Priya
Dept. of Information Technology
Thiagarajar College of Engineering
Madurai, Tamil Nadu

ABSTRACT
Today recruiters find their suitable human resources by searching in the job related web sites. In the same way, the job seekers also select their suitable jobs. The job seekers post their resumes in the multiple web sites. The HR tools available in the market make the job of recruiters easy by giving them the suitable resumes. Still, there is a good chance that the relevant documents may be missing in the list and unwanted document may exist in the list. This paper proposes a model for extracting resume information from different websites and makes the job of job recruiter easier by finding the suitable resume to fit their needs. Ontology is created with the suitable entities and their relationships for this domain. Each resume is split into four different sections namely – personal, education, skills and work experience. Attribute values are extracted from the resume documents. These values are updated in four different Resource Description Framework (RDF) files for each resume through ontology mapping. Resumes are ranked based on cosine similarity measure and then the ontologies are updated correspondingly. Here Graph database (i.e RDF), a NoSQL data model is used for storing the resumes and SPARQL is used for querying the documents. Experiments are carried out based on the retrieval time of relevant documents to find out the effective RDF model for storing the resumes.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval. I.2.7 [Artificial Intelligence]: Natural Language Processing – Text analysis.

General Terms
Documentation

Keywords
Big data, Information Retrieval, Semantic Web, Ontology, Resume Ontology, RDF, SPARQL

1. INTRODUCTION
The Big data analytics play a vital role in today's situation. Traditional Relational Database Management System (RDBMS) may not be a suitable data model for the large variety and vast amount of unstructured or semi-structured data. New analytical tools for processing large amount of unformatted data are coming up now-a-days.

But representation of data is more important than these. Many new data models say NoSQL data models [19] have come up to accommodate these large amounts of data. Appropriate data models have to be selected so that new knowledge can be discovered from the existing data by integrating them suitably.

Large amount of web contents are not machine understandable and readable, but have to be done only by human beings. Inferences can be easily made by human beings but not by machines. Retrieving information, ranking of documents, summarization, contextual search, etc becomes complex and tedious, since most of them require manual effort. Time consumption is more if one needs to refer or access too many documents to collect data and make summary on them.

Tim Berners Lee's vision was "To make the computers to think like human brains". The semantic web technologies overcome the above difficulties to a greater extent by adding domain based semantic information along with the current data. A core data representation format for semantic web is Resource Description Framework (RDF). RDF is a data model for web pages. RDF is a framework for representing information about resources in a graph form. It was primarily intended for representing metadata about WWW resources, such as the title, author, and modification date of a Web page, but it can be used for storing any other data. It is based on triples *subject-predicate-object* that form graph of data. All data in the semantic web use RDF as the primary representation language [14].

RDF Schema (RDFS) can be used to describe taxonomies of classes and properties and use them to create lightweight ontologies. Ontologies describe the conceptualization, the structure of the domain, which includes the domain model with possible restrictions. More detailed ontologies can be created with Web Ontology Language (OWL). It is syntactically embedded into RDF, so like RDFS, it provides additional standardized vocabulary. For querying RDF data as well as RDFS and OWL ontologies with knowledge bases, a Simple Protocol and RDF Query Language (SPARQL) is available. SPARQL is SQL-like language, but uses RDF triples and resources for both matching part of the query and for returning results of the query [15].

Semantic Web ontologies consist of taxonomy and a set of inference rules from which machines can make logical conclusions [12].

In this paper, we've proposed a graph data model for storing resume documents by developing an application using ontology mapping, thus providing the relevant resume information to the job recruiter by reducing manual effort. The proposed model uses the semantic web technologies like RDF to represent the information and SPARQL to retrieve information from it. Term vector based cosine similarity measure enables the ranking order for the resumes. This paper is organized as follows: Section 2 describes the related work, Section 3 explains the proposed methodology, Section 4 presents the implementation details that have been undertaken and Section 5 summarizes the work done with the future works explained.

2. RELATED WORK

Maryam Fazel-Zarandi et. al [1] proposed an ontology-based hybrid approach to effectively match job seekers skills by using the deductive model to determine the kind of match between a job seeker and skills required by recruiter. Ujjal Marjit et. al [2], proposed a technique for retrieving the resume information through the concept of Linked Data which enabled the web to share data among different data sources and to discover any kind of information. Koparappu of TCS Innovations lab [3], developed a system for automated resume information extraction to support rapid resume search and management which is capable of extracting several important informative fields from a free format resume using a set of natural language processing (NLP) techniques.

Zhi Xiang Jing et. al [4], presented a systematic solution of the information retrieval in online Chinese resume using the rule-based and statistical algorithm to extract information. Zhang Chuang et.al [5], had researched on resume document block analysis based on pattern matching, multi-level information identification and developed the biggest resume parser system. Celik et. al [6], developed a system to convert the resume into ontological structural model, thereby giving an efficient way for searching the Turkish and English resumes. Di Wu et. al [7], used ontology for the effective information extraction from the resumes using the WordNet for similarity calculation.

Salah T. Babekr et. al [8], developed a system which represents Web documents using the concept vector model and WordNet and done Personalization and text summarization for the documents. Zeeshan Ahmed et. al [9], propsed two different approaches to extract information like Rule Based Approach and Dependency Based Approach on the corpus of recipe documents. This kind of semantic annotation is useful for efficient answering of search queries, clustering, text summarization, etc. Mahendra Thakur et. al [10], developed a query based web search system by using the personal information of users by enabling the users to get the relevant web pages based on their selection from the domain list.

The literature survey also shows that the existing websites gives more advanced options for search criteria like location, keyword search, domain search, etc for easy document retrieval. But they have some disadvantages like - Linkedin website [20] retrieves the resumes which are mostly irrelevant; Resume Builder website [22] provides very minimum number of resumes; Indeed.com [21] gives some better results compared to other web sites; but it also gives some irrelevant resumes when compared to our proposed model. For example: when the keyword 'C' is used for search, all the resumes whose name has 'C' as the initial is also listed, whereas the actual requirement here is the programming language 'C'.

Our proposed model differs in the following ways – (i) Ontology mapping for categorizing the resumes (ii) Ranking of resumes based on term vector calculation. This hybrid model of storing the resume information retrieves the relevant resumes efficiently.

3. METHODOLOGY

The proposed model contains four main steps: (i) collection of resumes from various websites like indeed.com, linked in, etc. by

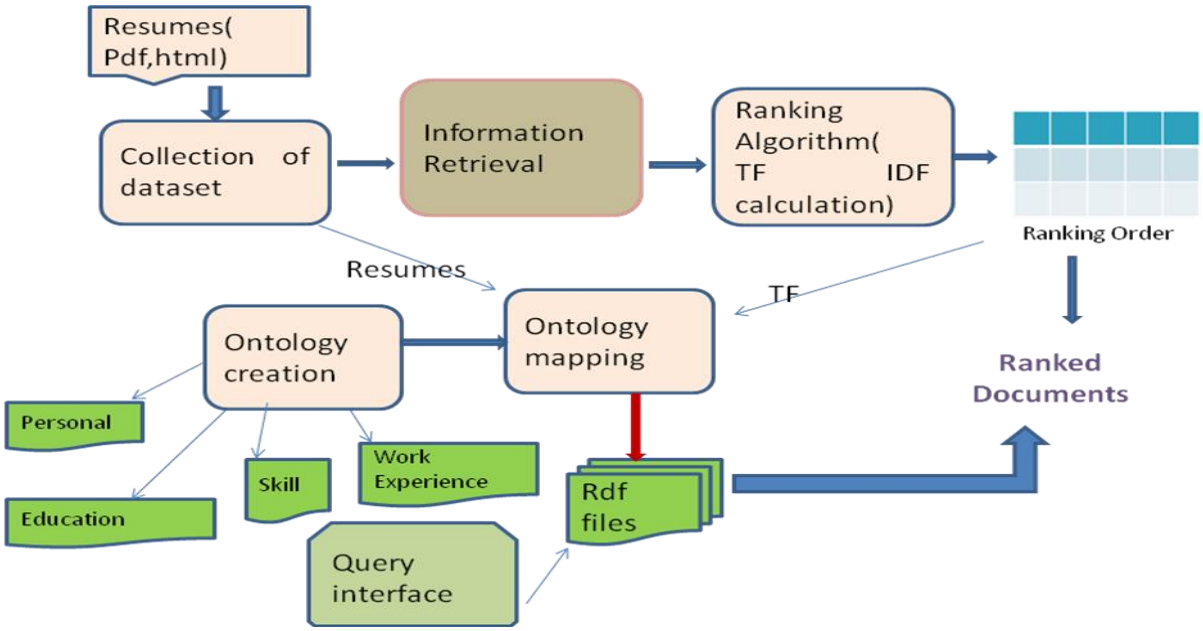

Figure 1 Model of Information Retrieval from Resumes

using appropriate keywords such as skill, experience; conversion of HTML or PDF or Doc format into common structured format (ii) domain ontology creation (iii) term vector calculation of each resume (iv) ontology mapping and ranking of each resume. The Figure 1 given below shows the proposed model and each step involved in the process are explained in the subsequent sections.

3.1 Collection of Datasets

The Resumes are collected from different websites like indeed.com, linked in, etc., by entering skill as a keyword. These websites have resumes which are uploaded by the job seeker to seek a job. The uploaded resumes are in different formats like PDF and HTML format. Retrieving and analysing data from HTML files is easier when compared to PDF files. So, PDF files are converted into HTML files using EasyPdfToHtml CONVERTER tool [18], an open source tool. The HTML files consist of resume information in unstructured format without any specific format, which is then converted into structured format.

3.2 Ontology Creation

Ontology is created based on the personal, education, skills and work experience details of resume with suitable vocabularies using Protégé tool [13]. A personal detail consists of the job seeker name, email id and address. Educational detail consists of completion of courses and percentage of marks in them, the institution in which they were studied.

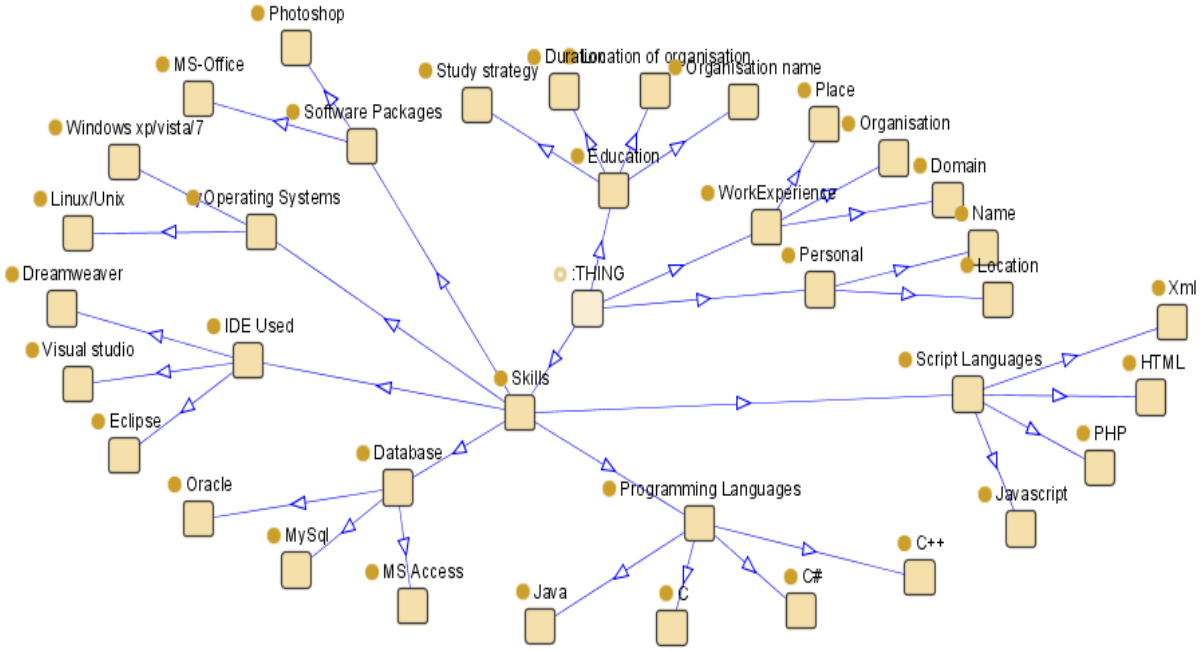

Figure 2 Ontology for resumes

In case of skills, it is categorized into many ways – (i) programming skills consists of the knowledge about programming languages like C, C++, Java, C#, JavaScript (ii) working experience in various operating systems like Windows 7, Windows 8, Linux, etc. (iii) database knowledge like DB2, Oracle, PostgreSQL, Sybase, etc. Each resumes is logically divided into four categories – personal, education, skills and work experience for easy information retrieval, thus 4 different RDF files for each resume. So for one resume we need 4 RDF files to store all the information.

3.2 Term Vector Calculation

TF-IDF is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document [24]. According to Garcia [11], the term vector calculation is given by,

TF = Term Frequency = number of times a term is repeated.
IDF = inverse document frequency = log ((N – n)/n)
n = number of documents containing a term.
N=Total number of documents.

Term weight is the product of Term Frequency and Inverse Document Frequency. It is denoted as w.

$$w = TF*IDF = TF*\log((N - n)/n)$$

Dot Product Calculation:
For two dimensions, x and y, the DOT Product is given by x1*x2 + y1*y2, where x1, y1 term weights and x2, y2 are IDF for corresponding documents.

Vector Magnitude Calculation:
For two dimensions, x and y, the vector magnitude is given by $\sqrt{(x1^2 + y1^2)}$.

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter [23].

document	C-TF	C++-TF	Java-TF	C#-TF	C-W	C++-W	Java-W	C#-W	c&&c++	c&&java
Aashima-K	1	0	0	0	-0.42488	0	-0	0	0.994059	0.98374
Abinash-K	1	0	1	0	-0.42488	0	-0.07756	0	0.994059	
Aditya-Dat	0	1	0	1	-0	0.04652	-0	0.728239	0.108839	#NUM!
Afsath-P.A	1	1	1	0	-0.42488	0.04652	-0.07756	0	1	
Akash-Dee	1	1	1	0	-0.42488	0.04652	-0.07756	0	1	
Amrish-Mis	1	1	2	1	-0.42488	0.04652	-0.15512	0.728239	1	0.9856
Ananth-Ch	3	1	4	0	-1.27465	0.04652	-0.31023	0	0.997368	0.99830
Ananth-Ch	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
Ankush-Ka	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
Aparna-Du	1	2	1	0	-0.42488	0.09304	-0.07756	0	0.994332	
Archana-In	2	2	2	3	-1.27465	0.18608	-0.38779	0	0.999355	0.9934
Ashish-Ku	1	0	0	0	-0.42488	0	-0	0	0.994059	0.98374
Avinash-M	1	0	1	1	-0.42488	0	-0.07756	0.728239	0.994059	
Balaji-Thod	3	2	0	0	-1.27465	0.09304	-0	0	0.999345	0.98374
Balakrishn	2	2	0	0	-0.84977	0.09304	-0	0	1	0.98374
Betma-Ind	0	1	0	0	-0	0.04652	-0	0	0.108839	#NUM!
Bhalchand	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
Bhushan-E	1	0	3	0	-0.42488	0	-0.23267	0	0.994059	0.9490
Kapil-Chav	0	0	2	10	-0.42488	0.04652	-0.07756	0	1	
butchibabu	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
C.-Ayyapp	1	0	0	0	-0.42488	0	-0	0	0.994059	0.98374
C.-Kolanch	1	0	0	0	-0.42488	0	-0	0	0.994059	0.98374
Chaitra-B.I	0	1	1	0	-0	0.04652	-0.07756	0	0.108839	0.17957
D.-Harithar	2	5	0	0	-0.84977	0.2326	-0	0	0.987525	0.98374
Danish-V-M	0	0	6	1	-0	0	-0.46535	0.728239	#NUM!	0.17957
Deepak-Ku	0	0	1	3	-0	0	-0.07756	2.184716	#NUM!	0.17957
Devanshi-A	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
DEVENDR	1	1	1	0	-0.42488	0.04652	-0.07756	0	1	
Dipti-Desh	2	3	3	1	-0.84977	0.13956	-0.23267	0.728239	0.998557	0.99624

Figure 3 Term vector calculations for each resume

Cosine Similarity is computed between query and each document by dividing the dot products by the product of the corresponding magnitudes. These term vector values are updated in OWL files using ontology mapping technique and thus resumes are ranked. For easy calculation, excel sheet is used and is shown in Figure 3.

3.3 Ontology Mapping

It is the technique by which the data from the document is mapped against its tag and RDF is updated. For example the following Figure 4 shows the resume containing the keyword "C#" 3 times.

Then the corresponding RDF files like experience and skills are updated with this information as shown below –

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE rdf:RDF <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:person="http://www.w3schools.com/rdf/" xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<rdf:Description rdf:about="D:/FINAL/experiment/input/Archana Indalkar.html">

<person:name>Archana Indalkar</person:name>
<person:skills>SKILLS</person:skills>
<person:skills>Languages: C, C++, Core Java, Advance Java (JSP, Servlet),C#.NET.</person:skills>
<person:skills>Operating Systems: DOS, Windows 98/ XP/ Linux.</person:skills>
<person:skills>Databases: MS Access, MySQL, SQL, PostGres SQL (Linux)</person:skills>

</rdf:Description>
</rdf:RDF>

personal

1. Archana Indalkar
2. (C#.NET) Software Application Developer
3. Pune, Maharashtra - Email me on Indeed: indeed.com/r/Archana-Indalkar

education

1. M. Sc.
2. Pune University - Pune, Maharashtra
3. May 2009
4. B.Sc.
5. Pune University - Pune, Maharashtra
6. May 2007
7. H.S.C
8. A.K. Jr. college - Pune, Maharashtra
9. May 2004
10. S.S.C
11. New English School Phursungi - Pune, Maharashtra
12. May 2002

skills

1. SKILLS
2. C, C++, Core Java, Advance Java (JSP),c#.NET,MYSQL
3. Languages: C, C++, Core Java, Advance Java (JSP, Servlet),C#.NET.
4. Operating Systems: DOS, Windows 98/ XP/ Linux.
5. Databases: MS Access, MySQL, SQL, PostGres SQL (Linux)

Figure 4 Original resume with keyword highlighted

Term frequency values are calculated and mapped with the corresponding document id.

<rdf:cs>
<rdf:name>Archana Indalkar</rdf:name>
<rdf:tf>
<rdf:cstf>3</rdf:cstf>
</rdf:tf>
</rdf:cs>

3.4 Query Interface

Job recruiter can query the required information by entering the skill as a keyword. SPARQL also enables the recruiter to view the personal, educational, skills and work experience separately for each resume. Jena APIs [16, 17] are used for using SPARQL and RDF. For example, the SPARQL query given below retrieves personal information of Aashima Khanna.

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "+
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> " +
"PREFIX person: <http://www.w3schools.com/rdf/>"+
"select ?"personal+
" where { "+" ?url rdfs:subClassOf <http://www.w3.org/2002/07/owl#ontology>.?uri rdfs:subClassOf ?url.?uri person:"personal" ?"+msg+".?uri person:name ?name.FILTER regex(?name ,\""Aashima Khanna"\",\"i\")}";

4. IMPLEMENTATION

Most of the web documents are unstructured or semi-structured. Structuring or formatting these data may take longer time and the need for these data is not for a longer period. So, they may not require RDBMS structure. Also they are available in plenty. In order to cater to the needs of Big Data analytics, RDF plays an important role in representing the unstructured/semi-structured data.

Java based application is developed to collect different resumes through web search and to convert into common structured format. The standard format can be obtained by the use of ontology. The resumes are ranked using the term vector calculation and ontology is updated as explained in the previous sections. JENA APIs [16, 17] are used for retrieving information from RDF. Figure 5 shows the snapshot of the part of the application, when a keyword is given to it.

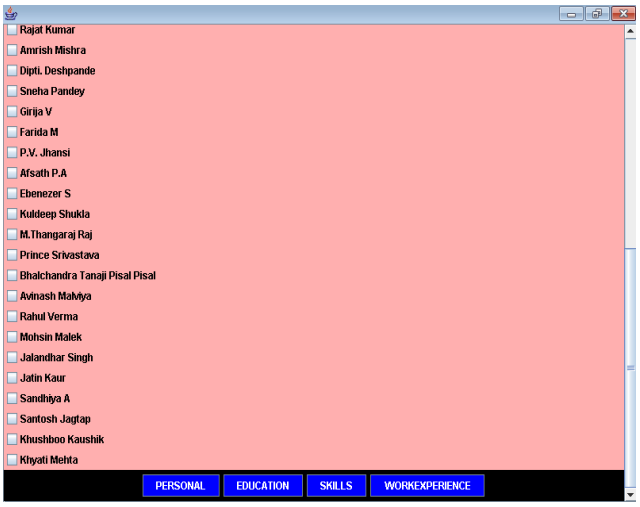

Figure 5 Relevant resumes with ranked order

When entering the keyword, the relevant resumes are obtained based on ranking order. Here the "Kapil Chawla" resume comes first because it has more term frequency values than others as shown in below Figure 6.

A	B	C	D	E	F	G	H	I	J	K
Bhalchand	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
Bhushan-E	1	0	3	0	-0.42488	0	-0.23267	0	0.994059	0.94909
Kapil-Chav	0	0	2	10	-0.42488	0.04652	-0.07756	0	1	1
butchibabu	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
C.-Ayyapp	1	0	0	0	-0.42488	0	-0	0	0.994059	0.983745
C.-Kolanch	1	0	0	0	-0.42488	0	-0	0	0.994059	0.983745
Chaitra-B.I	0	1	1	0	-0	0.04652	-0.07756	0	0.108839	0.179573
D.-Harithar	2	5	0	0	-0.84977	0.2326	-0	0	0.987525	0.983745
Danish-V-I	0	0	6	1	-0	0	-0.46535	0.728239	#NUM!	0.179573
Deepak-Ku	0	0	1	3	-0	0	-0.07756	2.184716	#NUM!	0.179573
Devanshi-A	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
DEVENDR	1	1	1	0	-0.42488	0.04652	-0.07756	0	1	1
Dipti.-Desh	2	3	3	1	-0.84977	0.13956	-0.23267	0.728239	0.998557	0.996243
Ebenezer-S	1	0	0	0	-0.42488	0	-0	0	0.994059	0.983745
Farida-M.h	1	1	0	1	-0.42488	0.04652	-0	0.728239	1	0.983745
Girija-V.htr	1	0	1	0	-0.42488	0	-0.07756	0	0.994059	1
Govind-Sur	1	1	1	0	-0.42488	0.04652	-0.07756	0	1	1
hameed-ha	0	0	0	0	-0	0	-0	0	#NUM!	#NUM!
Harika-A.h	0	0	4	0	-0	0	-0.31023	0	#NUM!	0.179573
Harshada ,	2	3	0	3	-0.84977	0.13956	-0	2.184716	0.998557	0.983745
Harshad-A	1	0	0	0	-0.42488	0	-0	0	0.994059	0.983745
Harshal-Ba	1	1	7	0	-0.42488	0.04652	-0.54291	0	1	0.747704
Heena-Kos	2	0	1	0	-0.84977	0	-0.07756	0	0.994059	0.995994
Hiral-Chud	1	0	0	0	-0.42488	0	-0	0	0.994059	0.983745
Indira-Naga	4	2	0	3	-1.69953	0.09304	-0	2.184716	0.998523	0.983745
Ipsita-Bhat	3	3	4	6	-1.27465	0.13956	-0.31023	4.369431	1	0.998307
Jalandhar-S	0	0	1	1	-0	0	-0.07756	0.728239	#NUM!	0.179573
Jaswandi-C	1	1	6	2	-0.42488	0.04652	-0.46535	1.456477	1	0.79592
Jatin-Kaur.	1	0	0	0	-0.42488	0	-0	0	0.994059	0.983745
Jignesh-Pr	0	0	2	2	-0	0	-0.15512	1.456477	#NUM!	0.179573

Figure 6 Comparing Term Frequency with retrieved resumes

4.1 Measuring Time

Experiments are carried out in three different ways to calculate the time required for retrieving the relevant resumes.

Case 1: Single RDF file for all resumes

Single RDF file means that personal, educational, skills and work experience details of a person are stored in a single RDF file and the time taken is measured. For example, if there are 10 resumes, all are stored in a single RDF file.

Case 2: Four RDF files for each resume

For each resume 4 RDF files are created to stores the details about personal, educational, skills, work experience separately and the time taken is measured. For example, if there are 10 resumes, the four sections are separated for each resume and all information is stored in four different RDF files.

Case 3: Three RDF files for each resume

For each resume, three RDF files are created. In that, two RDF files are used to store personal and education details. And another one file is used to store the skills and work experience details and the time taken is measured.

Table 1 Time taken when single RDF file is used

No of resumes	Single RDF Time taken (in ms)	Three RDF Time taken (in ms)	Four RDF Time taken (in ms)
50	2500	2300	2100
75	3000	2800	2600
100	4000	3500	3000

It is observed that (i) in case 3, if the recruiter wants to retrieve the personal or education details, it takes less time whereas recruiter wants to retrieve the details about skills or work experience, it takes some more time. (ii) in case 2, if the recruiter wants to know the personal details for a particular resume, the SPARQL query directly retrieves the personal details. It takes less time when compared to the above case. (iii) in case 1, it takes more time for searching. Comparing the above three cases, four RDF files for each resume is considered better, as it has less searching time than the others.

5. CONCLUSION

The proposed model collects resumes through web search and ranked based on cosine similarity measure. Ontology plays the vital role while extracting and keeping the information relevant and update. Thus the searching time for required information is minimal when information is stored in four different RDF files compared to other two models. The model also reduces the human effort required in seeking the relevant information.

In future, we plan to use Hadoop environment for text analytics purpose so as to speed up the whole process.

6. REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

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

[10] 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.

[11] Dr. Edel Garcia, “Term Vector Calculations A Fast Track Tutorial” published in November 3, 2005.

[12] Grigoris Antoniou and Frank van Harmelen, "A semantic Web Primer".

[13] http://protege.stanford.edu/

[14] http://www.w3schools.com/rdf/rdf_example.asp

[15] http://www.w3.org/TR/rdf-sparql-query/

[16] http://jena.sourceforge.net/tutorial/RDF_API/

[17] http://jena.apache.org/documentation/tdb/

[18] http://www.easypdftohtml.com

[19] http://www.systap.com/bigdata.htm

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

[21] www.indeed.com

[22] www.resume.com

[23] http://en.wikipedia.org/wiki/Cosine_similarity

[24] http://en.wikipedia.org/wiki/Tf-idf