Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 85 (2016) 915 – 921

Procedia Computer Science

International Conference on Computational Modeling and Security (CMS 2016)

Concept Based Dynamic Ontology Creation for Job Recommendation System

Uma Pavan Kumar Kethavarapu [a] Dr.S.Saraswathi[b]

[a]*Research Scholar,CSE Department,Ponicherry Engineering College.,Pondicherry-605001, India*
[b]*Professor and HOD IT Department, Pondicherry Engineering College, Pondicherry,-605001, India*

Abstract

The basis of our research is to construct a job recommendation system to the job seekers by collecting the job portals data. Due to huge amounts of the data in job portals the employers are facing difficulty in the identification of right candidate for the required skill and experience. The job seekers are also facing the problem of getting the suitability of the job based on their skill and experience. The knowledge acquisition based on the requirements is very difficult in case of huge amounts of the data sources. In fact classical development of domain ontology is typically entirely based on strong human participation. It does not adequately fit new applications requirements, because they need a more dynamic ontology and the possibility to manage a considerable quantity of concepts that human cannot achieve alone. The main focus of our work is to generate a job recommendation system with the details of job by taking account into the data posted in the web sites and data from the job seekers by the creation of dynamic ontology. We strongly believe that our system will give the best outcome in case of suitable job recommendation for both employers and job seekers without spending much time. To achieve this first we have extracted the data from various web pages and stored the collected data into .csv files. In the second stage the stored input files are used by the similarity measure and ontology creation module by generating the corresponding Web Ontology Language (.owl) file. The third stage is creating the ontology with the generated .owl by using protégé tool .

Peer-review under responsibility of the Organizing Committee of CMS 2016

Keywords:Ontology;Job Portals;Job Seekers;Web Ontology Language;ElasticSerach

* Uma Pavan Kumar Kethavarapu. Tel.:+91-9844068669.
*E-mail address:*umapavanmtech@gmail.com

Peer-review under responsibility of the Organizing Committee of CMS 2016
doi:10.1016/j.procs.2016.05.282

1. Introduction

Ontology is a formal explicit specification of shared conceptualization which will provide Common semantics for agent communication. Currently, applications mostly exchange information on the basis of passing parameters or data, formatted according to pre-defined strict syntaxes this approach is known as the exactness method [1]. This method has the advantage of allowing total error management, except application bugs of course, but leaves no space for data interpretation. In consequence, reasoning on data of this type is virtually impossible because of the limits of its definition. Ontologies provide a richer knowledge representation that improves machine interpretation of data. For this they become to be widely used in information systems and applications, and ontology construction has been addressed in several research activities. Ontology must be able to grow dynamically without deviation of the existing applications. At the same time computational time for discovering the best matches between several ontologies is expensive, therefore the technique must maintain previous discovered alignments and common usages in order to quickly recognize similarities between concepts and to compute only new information. Ontology is designed not only to provide a complete view of domain concepts but also to identify quickly and accurately similarities between concepts, even if not identical, and to conduct consistent alignments [2]. The method we are working on gives the answer for the statement that is within the specified domain without taking much time will give the best matching of the job recommendations by considering the previous details and similarities of the key skills or work locations of the job seekers.

Ontologies can be built as manually, semi-automatically or automatically. If ontologies are constructed manually then we can term that ontology as manual ontology, while constructing the ontologies if human intervention is required then we can name that as semi-automatic ontologies, if the system takes care about complete construction of ontologies then we can say that as automatic ontology. A collection of documents that are related to the user's request are produced by comparing the user's request with an automatically generated index of the textual content of the documents present in the system by means of a computerized process, called document Retrieval (also known as Information Retrieval) [3,4]. Text retrieval in which information is basically stored in the form of text is a branch of information retrieval.

In some cases there may be a chance of changing the requirements in the input to the system. In such cases we must use the dynamic ontologies so as to embed the changes in the existing result sets. The organization of the paper is as follows in the section 2 we are going to explain the methods available in automatic ontology construction. In section 3 the details about various automatic ontology comparisons, in section 4 the explanation about the job recommendation system that we are constructed in section 5 future scope and conclusion is explained.

2. Automatic Ontology Construction

The problem with semi-automatic ontology construction is all the available methods are not suitable to the exact requirements and all they are domain-specific in nature [5]. So to address these issues the automatic ontology construction is introduced. The automatic ontology construction can be done based on the following context

Conversion of Ontologies as per requirements

XML to OWL

UML to OWL

Generating ontology from an annotated business model.

Mining Based

The following are various methods available in the literature [6, 7] for automatic ontology construction Based on Mining.

SALT (Standardization of Lexical and terminological resources):

Based on the lexical and terminal analysis on textual data with customer contract data.

Creating concurrent semantic annotations for PDF documents.

Limitations: Static approach, User intervention is required as it generates more concepts than required.

TERMINAE:

Used to build automatic ontology from text.

A computer aided knowledge engineering tool written in java.
Also used to create a new operating system from scratch.
Limitation: We can’t insert new topic text after creation of ontology.

Learning OWL ontology from free text:

Automatic generation of ontologies based on the analysis of texts followed by the use of Word Net [8].
Limitation: Suitable to more general reference knowledge.
Artequakt:
Dynamically link a knowledge extraction tool to achieve continuous support to the extraction mechanism [9].
Consider the domain of artists and paintings.
Extracts knowledge about the artists from web by taking name of artists as input.
Limitation: Extracting similar or overlapping data from various sources.

External Knowledge Based

Design of the Automatic Ontology Building System about the Specific Domain Knowledge [9]
Domain-Specific Knowledge Acquisition and Classification Using WordNet
Enriching Very Large Ontologies Using the WWW.

3. Tools Of Automatic Ontology Construction

In the construction of Automatic ontology five parameters [10, 11, 12] are playing a vital role, those are Extraction, Analysis, Generation, Validation and Evolution. The following table gives the analysis of various tools based on the above mentioned parameters.

Table 1. Comparative Analysis of Automatic Ontology construction Tools .

Method	Extraction	Analysis	Generation	Validation	Evolution
Generating an ontology from an annotated business model	Human	-	Direct transformation using XSLT files.	Human	-
XML2OWL	Static table of correspondences	-	Direct transformation using XSLT files.	Human	
UML2OWL	Using Semi-automatic approach	-	Direct transformation using XSLT files.	Human	
TERMINAE	NLP techniques	Concept relationships analysis	No standard ontology representation	Human	-
SALT	NLP techniques. Multi entries.	Similarity analysis of concepts	No standard ontology representation	–Limited human intervention	-
A new Method for Ontology Merging based on Concept using WordNet	-	Based on semi-automatic ontology	Automatic merging. No standard ontology representation.		-
Design of the Automatic Ontology Building System about the Specific Domain Knowledge	Main concept defined by a domain expert.	-	Human Intervention		-
Enriching Very Large Ontologies Using the WWW	Enrich existing ontology	-	Human Intervention		-

4. Proposed Research

Based on the above analysis we are going to construct one automatic ontology with similarity measure threshold value and re ranking method. The process we have adopted for the construction of automatic ontology for the job recommendation system is as follows.

Initially, the web log data can be collected from the Job portals. After the raw data can be collected, the pre-processing can be done. Initially, ontology creation and mapping will be done by analysing various properties of ontologies in order to deduce alternate semantics that may apply to other ontologies, and therefore create a mapping. The feature extraction module based on TF-IDF similarity, and then Indexing and ranking of information by Rabin Fingerprint algorithm and ranking can be done by semantic similarity measure [13]. Once the ontology is mapped, the relevant information will be retrieved effectively based on the user query. That is for the input query keyword, matching will be performed with the mapped ontology and the exact information's will be retrieved using the computed similarity score. Lastly, by means of the matching result the related report were generated from the document repository. The implementation is done in Java with protégé [14, 15] software tool for ontology creation and updating. The performance of the proposed system will be analysed in terms of accuracy, recall and F-measure.

Extraction: The source of extracting the data is web pages and we are using the domain of job portals such as naukri, monster and times job. The procedure here is we will input URL name the web page and the Read Data procedure will generates the .csv file. The same procedure will be repeated to multiple web pages so as to generate the .csv files. All the generated files will be collected in an input directory.

Analysis: In our system the analysis part is pre-processing such as identification of the URL information based on the requirement like skill, experience and qualification, actually this step is the basis for the generation of the .csv files forms the web portals.

Generation: The generation of the .csv files and after that by using re-ranking and similarity measures we are generating the .owl file which is the source file to the ontology construction, after populating the .csv files based on those files the Run() method integrates the multiple .csv files and filters the required records with similarity measure and re-ranking approaches. Once this process is done then the Run () will generates the newtest.owl file. Based on the .owl file we can automatically construct the ontology with the specified attributes of the job recommendation system.

Validation: The validation process is done manually by the user after generation of the ontology by using protégé tool.

Evolution: The evolution process is by default property in our system reason is that as we are using the streaming data in the form of web portals the information is up to date and dynamic so based on that only the generation of .csv and .owl is done, with the generated .owl we are generating the ontology with protégé tool.

Accuracy: In addition to the above parameters we are considering the one more parameter that is accuracy of the result. In our implementation we are using similarity measure along with threshold value by comparing the word net source of the data as a dictionary for searching the words, will give some value based on that we can generate the records as per our requirements.

The following functionalities we have used in the implementation [16 17 18].

FindMeasure (Query, data, originaldata)-Gives the measure value of the given data by comparing with the original data.

SelectBest (data [], score, originaldata)-Gives the best score for the given data in the query from the original data.

Score Value: The score value is the deciding factor for the generation of relevant records and this will works based on the user given threshold value like score>0.1

Some of the mathematical computations performed in our implementation of work.

tf (t in d) correlates to the term's frequency, defined as the number of times term t appears in the currently scored document d [19].

$$tf(tind) = frequency^{1/2} \qquad (1)$$

idf(t) stands for Inverse Document Frequency. This value correlates to the inverse of docFreq (the number of documents in which the term t appears). This means rarer terms give higher contribution to the total score. idf (t) appears for t in both the query and the document; hence it is squared in the equation [20].

$$idf(t) = 1 + \log\left[\frac{numDOCS}{docFreq+1}\right] \quad (2)$$

In the measuring of the relevancy of the records populated by the .csv files we are using score value of the records generated in such a way that , a threshold value we will mention to get more relevant records with similarity measure. Here the measure will compute the score value out of the existing records and their attributes with the help of word net repository [14, 15] and it outputs the most relevant records with the help of that only the .owl will be generated.

The data set considered for the research is web portal data such as naukri, monster and times jobs etc., from these portals we are generating the .csv and the system configuration is HP with 8GB RAM and Core I3 processor with Net Beans 8.1.The other soft wares required to test the outcome are Word Net 2.1 and protégé 4.1.The further research of our work requires Python, MongoDB along with elasticsearch so as to get the global job recommendation systems.

The term frequency in our research is used to get the number of times a term occurred in a document. For example "Java" is a term which is occurred 158 times, " Python" is term which occurred 185 times and the term "Hadoop" occurred 152 times, machine learning occurred 3 times. We have used the terms like "Python" "Bangalore" ,"Java", "Chennai" and "Hadoop" ,"Mumbai" together so as to get the resultant records.

Another computing we have done is inverse document frequency where we can get the score value of the term in a document. For example if I am giving java then the idf value is 0.5235 and similarly for Python I am getting the value of 0.85732, where as in case of machine learning it is like 0.03010.Based on the score value we are going to compare with minimum threshold so as to get the more relevancy of the records. The threshold value we have taken in the experiment is 0.01 and we can see the outcome of the query based on the input query along with the threshold value.

Table2. Computation of the term relevancy score for the Input Query

Input Query	Term frequency Value	IDF Value	Remark
"Python Bangalore"	[152 38]	0.0577	Record accepted
"Java Chennai"	[185 49]	0.0965	Record accepted
"Hadoop Mumbai"	[122 34]	0.0414	Record accepted
"Tibco" "Hyderabad"	[10 8]	0.0080	Record rejected
"Python Chennai"	[152 18]	0.0273	Record accepted
"Java Hyderabad"	[185 69]	0.1267	Record accepted
"Hadoop Noida"	[122 24]	0.2928	Record accepted
"Tibco" " Pune"	[10 18]	0.0018	Record rejected

The following is the outcome of the similarity measure in the implementation of proposed research. Once the generation of .owl is over the next step is to generate the ontology with set of attributes and the relationships. Initially the entire data items are converted into ontology format and there after we can query the ontology based on the requirements. In the following diagram we can see the attributes and relationships surrounded by ontology and the export of ontograph is possible for the query where workplace is equal to “Bangalore” .The following is the sample diagram of OntoGraph generated for the query where work place=”Bangalore”.

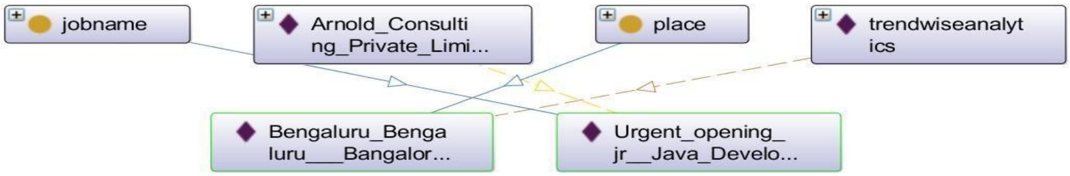

Fig. 1. Ontology Graph With .OWL using Protégé Tool.

The following is the outcome of the similarity measure in the implementation of proposed research. Once the generation of .owl is over the next step is to generate the ontology with set of attributes and the relationships. Initially the entire data items are converted into ontology format and there after we can query the ontology based on the requirements.

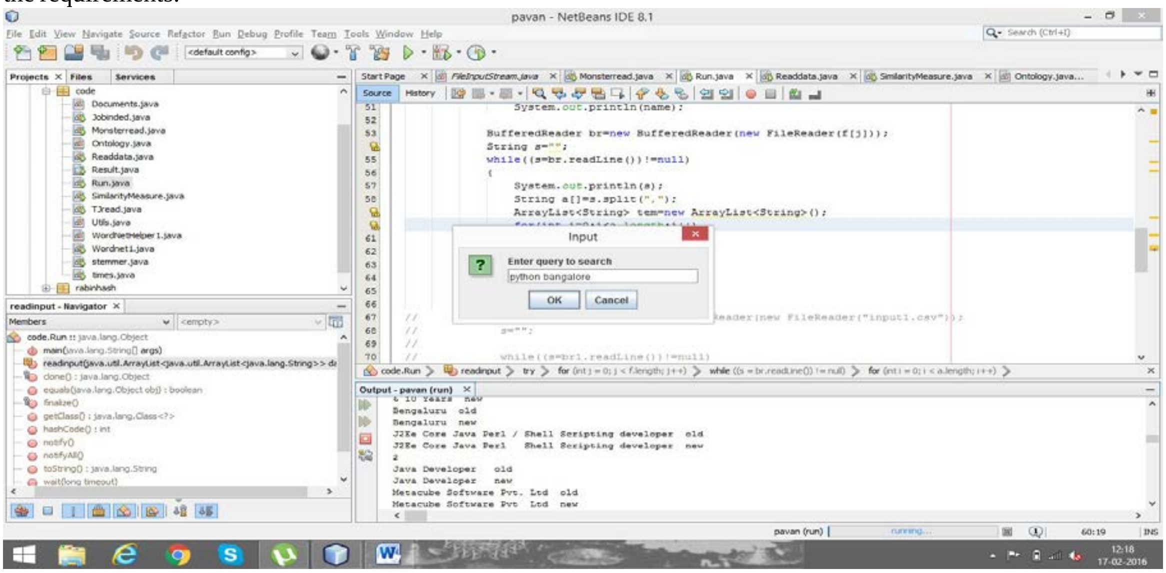

Fig. 2.Similarity Measure computing based on the given query Python Bangalore.

The following is exported version of the DOT file from the created ontology.

digraph g {

"trendwiseanalytics" -> "Bengaluru_Bengaluru___Bangalore" [label="in_place"]

"place" -> "Bengaluru_Bengaluru___Bangalore" [label="has individual"]

"jobname" -> "Urgent_opening_jr Java_Developer_bangalore" [label="has individual"]

"Arnold_Consulting_Private_Limited" -> "Urgent_opening_jr Java_Developer_bangalore" [label="has_vacancy"]

}

5. Future Scope and Conclusion

The overall theme of our work is to construct the automatic ontology with in less time and more accurate. The domain we are working is job recommendation system through which we are taking the input as web pages and generation of the .csv files. In the next stage these multiple .csv's are used to generate the .owl file. And in the generation of ontology we are giving some query so as to output the records. At this stage we are using similarity measure based on the term frequency and inverse document frequency which is helpful in the generation of content with more relevant data. The generated ontology is competent on par with manual ontology in the context of accuracy and similarity of the data generation. And the benefits of our work compared with manual ontology construction is time saving and complete automation of tasks like .csv generation , getting the .owl from multiple .csv's and finally generation of automatic ontology. The elastic search [21] and integration other social media sites information to the job seekers data is our next focus .The expected outcome is to filter the irrelevant and fake resumes if any conflicts or any negative results we are getting. The methodology we are going to adopt is exhaustive search [22] on the social sites along with the job portals so that one global job recommendation system of notifications generation for both the employers and job seekers.

References

1. Nelia Lasierra,"Designing an Architecture for Monitoring Patients at Home: Ontologies and Web services for Clinical and Technical Management Integration," IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 18, NO. 3, MAY 2014 p 10-24.
2. Mark Roantree, Automating the Integration of Clinical Studies into Medical Ontologies", 2014 47th Hawaii International Conference on System Science p 190-205.
3. Vishal Jain, "Architecture Model for Communication between Multi Agent Systems with Ontology", International Journal of Advanced Research in Computer Science, Volume 4, No. 8, May-June 2013 p 123-134.
4. Pavel A. Smirnov", Domain Ontologies Integration for Virtual Modelling and Simulation Environments", Procedia Computer Science, Elsevier, Volume 29, 2014, Pages 2507–2514.
5. S.MONISHA," A FRAMEWORK FOR ONTOLOGY BASED LINK ANALYSISFOR WEB MINING", Journal of Theoretical and Applied Information Technology, 20 thMarch 2015. Vol.73,No.2 p 32-45.
6. Vishal Jain," MINING IN ONTOLOGY WITH MULTI AGENT SYSTEM IN SEMANTIC WEB: A NOVEL APPROACH", The International Journal of Multimedia &Its applications (IJMA) Vol.6, No.5, October 2014 45-55.
7. Fernández López, M.," Overview Of Methodologies For Building Ontologies", Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)Stockholm, Sweden, August 2, 2013.
8. Pratibha Gokhale," Ontology Development Methods", DESIDOC Journal of Library & Information Technology, Vol. 31, No. 2, March 2011, pp. 77-83.
9. Ying Ding", Ontology Research and Development – A Review of Ontology Mapping and Evolving", Journal of Information Science, 28(5), 375-388, 2012.
10. Helena Sofia Pinto", A Methodology for Ontology Integration", K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada, Copyright 2001 ACM 1-58113-380-4.
11. Trong Hai Duong," Complexity Analysis of Ontology Integration methodologies: a Comparative Study", Journal of Universal Computer Science, vol. 15, no. 4 (2009) p 18-25
12. Mark Roantree", Automating the Integration of Clinical Studies into Medical Ontologies", IEEE Computer Society, 2014.
13. YuxiaHuang", Using Ontologies and formal concept analysis to integrate heterogeneous tourism information ", IEEE Transactions in Emerging Topics in Computing, June 2015 p 25-33.
14. O. Iroju," State-of-the Art: A Comparative Analysis of Ontology Matching Systems ", African Journal of Computing &ICT, Vol 5. No. 4, June 2012 ,IEEE ISSN 2006-1781.
15.Yuxia Huang and Lingbian Using Ontologies and formal concept analysis to integrate heterogeneous tourism information, IEEE Transactions in Emerging Topics in Computing, June 2015 p 45-52.
16.Lihua ZHAO, Ontology Integration for Linked Data Springer, J Data Semant (2014) p 101-109.
17.Lihua ZHAO, Integrating Ontologies Using Ontology Learning Approach,IEICE Transactions, vol. E-96, no. 1 January,(2013).
18.Mark Roantree Automating the Integration of Clinical Studies into Medical Ontologies, IEEEComputer Society, 2014 p 34-46.
19.Trong Hai Duong, Complexity Analysis of Ontology Integration Methodologies: a Comparative Study, Journal of Universal Computer Science, vol. 15, no. 4 (2009), p.877-897
20.Dr.S. Saraswathi and S. Venkataramanasam Design of dynamically updated automatic ontology for mobile phone information retrieval system Int. J. Metadata, Semantics and Ontologies, Vol. 9, No. 3, 2014.
21.Harith Alani, Web based Knowledge Extraction and Consolidation for Automatic Ontology Instantiation, ACM, October 23-25, 2013 p165-172.
22.Isabelle Guyon, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 2013 , p.1157-1182.