Asian Journal of Information Technology 5 (11): 1197-1201, 2006

An Ontology Based Approach for Information Retrieval and Visualization

[1]C.R. Balamurugan and [2]G.V.Uma
[1]Department of Computer Science and Engineering, Jerusalem College of Engineering, Chennai-601302, Tamilnadu, India
[2]Department of Computer Science and Engineering, College of Engineering, Anna University, Chennai-600 025, Tamilnadu, India

Abstract: In study to be able to exchange the semantics of information, one first needs to agree on how to explicitly model it. Ontologies are a mechanism for representing such formal and shared domain descriptions. They can be used to annotate data with labels (metadata) indicating their meaning, thereby making their semantics explicit and machine-accessible. For Visualizing the retrieved results, schema visualization techniques are used, which primarily focuses on the structure of the ontology, i.e. its concepts and their relationships. Unlike data models, the fundamental asset of ontologies is their relative independence of particular applications, i.e. ontology consists of relatively generic knowledge that can be reused by different kinds of applications/tasks. In general, for information retrieval, though various techniques are evolved, effective and efficient retrieval mechanisms lack in providing results in a better way. Hence this paper aims at providing a classical ontology based approach for information retrieval and visualization.

Key words: Ontology, information retrieval, RDF, XML, visualization, gate, POS tagger, chunker

INTRODUCTION

The traditional solution to the problem of recall and precision in information retrieval employs keyword-based search techniques[1]. Documents are only retrieved if they contain keywords specified by the user. However, many documents contain the desired semantic information[1], even though they do not contain user specified key words. In study to overcome the shortcomings of key word-based technique[1] for information selection a concept-based model using ontology[2] is proposed. Ontology is a collection of concepts and their interrelationships, which can collectively provide an abstract view of an application domain. It is important to insure that high precision and high recall will be preserved during concept selection for documents or user requests, in conventional key word search[1] the connection through the use of ontology between keywords and concepts selected from documents to be accessed for retrieval is carried out manually a process.

The above Fig. 1 shows the general architecture for the proposed Ontology based Information Retrieval System.

RESOURCE DESCRIPTION FRAMEWORK

The Resource Description Framework (RDF)[3,4] is an infrastructure that enables the encoding, exchange and reuse of structured metadata. RDF[3,4] is an application of XML[4] that imposes needed structural constraints to provide unambiguous methods of expressing semantics. The structural constraints RDF imposes to support the consistent encoding and exchange of standardized metadata provides for the interchangeability of separate packages of metadata defined by different resource description communities. RDF additionally provides a means for publishing both human-readable and machine-processable vocabularies. Vocabularies are the set of properties, or metadata elements, defined by resource description communities.

The following Fig. 2 depicts the organizational Structure of RDF elements

ONTOLOGY

An ontology is a formal, explicit specification of a shared conceptualization. It contains a set of distinct and identified concepts related by a set of relations. It provides a shared and common understanding of a

Corresponding Author: C.R. Balamurugan, Department of Computer Science and Engineering, Jerusalem College of Engineering, Chennai - 601302, Tamilnadu , India

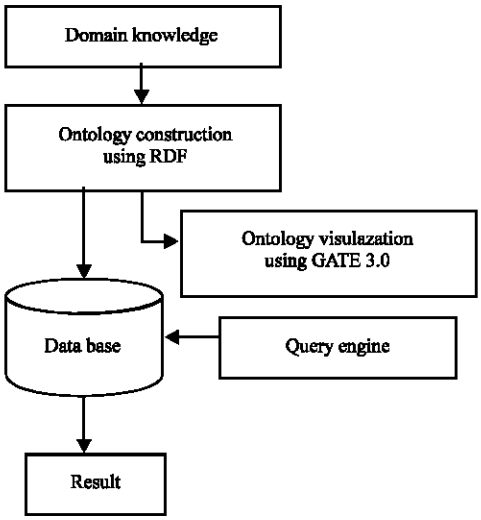

Fig. 1: General architecture of the system

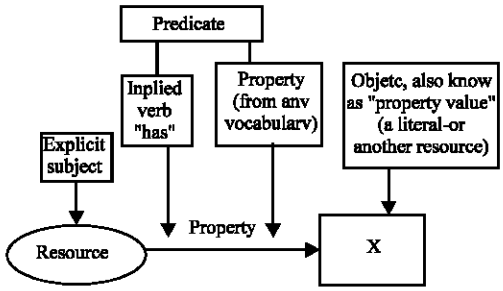

Fig. 2: Organizational structure of RDF elements

domain that can be communicated between people and heterogeneous and widely spread application systems. Ontology must be a shared and consensual terminology because it is used for information sharing and exchange.

There are roughly four kinds of ontologies: document ontologies, metadata ontologies, domain ontologies[5] and service ontologies. Our system is based on the third category named domain ontology. Components of different sports has been identified and marked up by Resource Description Framework, which gives the ontology.

VISUALIZATION

An ancient proverb says "A picture says more than a thousand words". Visualization is used to create and manipulate a graphic representation from a set of data. Some techniques will be appropriate only for specific applications while others are more generic and can be used in many applications. In our system the constructed ontologies are visualized using the tool GATE 3.0[6]. Visualized ontology gives an abstract view of the domain. The user can easily retrieve and visualize the information about the domain from the visualized ontology.

Gate: GATE stands for General Architecture for Text Engineering. GATE is an infrastructure for developing and deploying software components that process human language. GATE helps scientists and developers in three ways:

- by specifiying an architecture, or organisational structure, for language processing software;
- by providing a framework, or class library, that implements the architecture and can be used to embed language processing capabilities in diverse applications;
- by providing a development environment built on top of the framework made up of convenient graphical tools for developing components.

GATE can be thought of as a Software Architecture for Language Engineering 'Software Architecture' is used rather loosely here to mean computer infrastructure for software development, including development environments and frameworks, as well as the more usual use of the term to denote a macro-level organisational structure for software systems Language Engineering (LE) may be defined as: The discipline or act of engineering software systems that perform tasks involving processing human language. Both the construction process and its outputs are measurable and predictable. The literature of the field relates to both application of relevant scientific results and a body of practice.

The Ontogazetteer or Hierarchical Gazetteer, is an interface which makes ontologies to be visualized in GATE[6], supporting basic methods for hierarchy management and traversal. In GATE[6], an ontology is represented at the same level as a document and has nodes called classes. The OntoGazetteer assigns classes rather than major or minor types and is aware of mapping between lists and class Ids. There are two Visual Resources, one for editing the standard gazetteer lists and one for editing the ontology itself.

SYSTEM DESIGN

The following Fig. 3 shows the detailed design of the developed system.

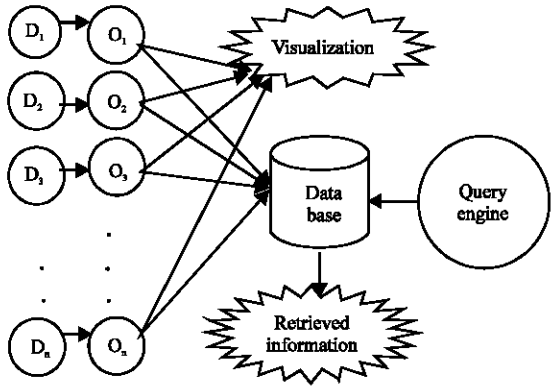

Fig. 3: Detailed design of the system

IMPLEMENTATION

With the help of the domain knowledge, concepts like class, subclass, property, domain and range for different domains (Cricket, Hockey, Football and Music) has been identified. Concepts are nothing but ontological components. Using the above identified components ontologies for different domains such as Cricket[7], Football, Hockey[8] and Music has been constructed using Resource Description Framework. Constructed Ontologies are then visualized using the tool GATE 3.0[6] (General Architecture for Text Engineering). The visualized output will give an abstract view of the domain to the user. The user can get the information about the domain through this visualized ontology. If the size of the ontology is large, then the possibility to get the required information from the ontology using this visualized output is less. In order to meet the user's requirement, an effective query engine is developed with a centralized database.

The constructed ontologies mainly focus on two ontological components. One is 'Class' and another one is 'Property'. Each class has 'n' number of 'subclasses'. Also each subclass has their subclasses and so on. There is no limit for a class to have the number of subclasses. When a property is defined, there are a number of ways to restrict the relation. Normally domains specify a property and ranges can be specified. The property can be defined to be a specialization of an existing property. The property defines about the relationship between two or more classes. Domain in the property gives the classes involves in a relation. Range gives the possible values of the relation.

For an effective retrieval system, there must be a database to store the ontologies. And the database must be centralized and store the ontologies irrespective of the domain to be stored. Initially all the tables in the database are empty. During run time, when a user select one domain, the ontology for the corresponding domain will be stored in the database. The database designed for this system has three tables. One to store the information about the first ontological component 'class' and rest of the two tables are used to store the information about the second ontological components 'property. So a database with three tables with relevant fields is created to store the ontology into the database using ORACLE 9i.

A User Interface is developed using JAVA to make the user to interact with the system. It allows the user to opt the domain among the set of given domains, for which the user is interested to know the information. When the user opts the domain, two things will be automatically done in the system. First one is, the ontology for the corresponding domain will be stored into the database[9]. The system uses the algorithm mentioned in the next section (5.1) to store the ontology in to the database. Secondly it invokes the process of query engine. There is two parts in the query engine. First one is the system will provide the set of predefined questions (query) corresponding to the domain of interest on the screen. Among the set of queries on the screen, user can select one for which the answer is needed. Now the information will be retired from the database and displayed on the screen. Based on the user need, the predefined queries would be updated. Second process in the query engine mechanism is the user can give his/her own query using Natural Language Processing Query Engine. Here the system will accept the query given by the user and parsed using POS tagger. The POS Tagger annotate each and every word in the query with the corresponding tag. Then this annotated text will be given as the input the noun phrse Chunker. This noun phase chunker will identifies all the nouns available in the query and it will be tagged. The output of this parse gives the noun phrases available in the system. Then all the information corresponding to the identified nouns will be retrieved from the database and displayed on the screen.

Algorithm: The following algorithm stores the ontology, which is constructed using RDF into the database.

Step 1: Store the whole RDF as a string (S).

Step 2: Generate tokens using string tokenizer by passing the string (S) and the delimiters < and > as parameters.

Step 3: Fetch each tokens and check the type of the token

Step 4: If the type='rdfs:label' then ID for that token is extracted and stored in to the table 'class' for the field 'class'.

Fig. 4: Visualization of cricket ontology

Fig. 5: Information retrieval for cricket domain

Step 5: If the type='rdfs:subclassof' then resource for the token is extracted and stored in to the table 'class' for the 'subclassof' field.

Step 6: If type='rdfs:comment' then the comment is stored in to the table 'class' for the field 'description'.

Step 7: if type='rdf:property' then the property name is stored in the string 'p'.

Step 8: if type='rdfs:domain' then a tuple in the 'domain' table will be filled with the corresponding resource name as 'domain' field and value of 'p' as 'property' field.

Step 9: if type='rdfs:range' then a tuple will be in the 'range' table filled with the corresponding resource name as 'range' field and value of 'p' as 'property' field.

The query engine uses this database for information retrieval.

RESULTS AND DISCUSSION

The following screenshots show the output of the system. Figure 4 shows how the cricket ontology is utilized using GATE and Fig. 5 shows the output of the information retrieval process. Like cricket ontology the system will support to visualize each and every ontologies irrespective of the domain and information retrieval can also takes place.

As per the definition of Ontology this visualized ontology gives an abstract view of the domain. The user can also use the natural language query processor to retrieve the information. Figure 5 shows the output of the system which responding the query "List the names of the teams played football so far" in an efficient manner.

CONCLUSION

An ontology based visualization and information retrieval system is developed. The main advantage of this system is, it uses single database with three tables to store ontologies of different domain. There is no need to have different database for different ontologies. The database is designed in an efficient manner in which the information in the ontology will be stored during run time. Constructed ontologies have been effectively visualized using the tool GATE. Visualized ontology itself gives the abstract view of the domain. It uses a query engine to retrieve information, which has a set of predefined queries. In addition with the predefined query processing mechanism, a natural language query processing mechanism also developed. Using this the user can give their own query in the natural language text English. The query processor is deigned in such a manner that the it will identify noun phrases available in the query and retrieved the information for the corresponding noun phrases from the table and it will be displayed in the screen.

Future work: In future, in addition with the existing ontology more number of properties can be added to optimize the size of the ontology. Then the constructed ontology can be measured with various measures and attributes such as reusability, adaptability, portability and reliability.

REFERENCES

1. Lazic, J.L. and S. Sanja, 2000. Hrvoje Stancic Information Retrieval Techniques at http://www.carnet.hr/CUC/cuc2000/radovi/prezentacije/F/F1/F1_f. pdf
2. Khan, L. and D. McLeod, 2000. Audio Structuring and Personalized Retrieval Using Ontologies, in Proc. of IEEE Advances in Digital Libraries, Library of Congress, Bethesda, MD, pp: 116-126.
3. Santtu Toivonen, 2001. Using RDF(S) to Provide Multiple Views into a Single Ontology, In Proceedings of the Semantic Web Workshop Hongkong, China.
4. Mike Uschold and Michael Gr¨uninger, 1996. Ontologies: Principles, methods and applications. Knowledge Engineering Review, 11: 93-155.
5. Wache, H., T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hubner, 2001. Ontology-Based Integration of Information – a Survey of Existing Approaches. In Stuckenschmidt, H. (Edn.). IJCAI-01 Workshop: Ontologies and Information Sharing, pp: 108-117.
6. Cunningham, H., D. Maynard, K. Bontcheva and V. Tablan, 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics.
7. http://www.fihockey.org/vsite/vorg/ page/ home/
8. Decker, S., F. Van Harmlen, J. Broekstra, M. Erdmann, D. Fensel, I. Horrocks, M. Klein and S. Melnik, 2000. The Semantic Web-on the respective roles of XML and RDF. IEEE Internet Computing.
9. Fabiane Bizinella Nardon, 2003. Lincoln de Assis Moura Jr, Beatriz de Faria Leao, Using RDF and deductive Database for knowledge sharing in healthcare, in http://www.tridedalo.com.br/ publications/ iswc2003.pdf.