RESEARCH ARTICLE OPEN ACCESS

Methodology for Ontology Mapping On Semantic Web

Lakshmi . S
Assistant Professor
Department of Computer Science
Sri Adi Chunchanagiri Women's College, Cumbum
Tamilnadu - India.

ABSTRACT

Ontology mapping is important processes in ontology engineering. It is obligatory by the decentralized nature of both the WWW and the Semantic Web.Ontology mapping can be used to establish efficient information sharing by determining correspondences among such ontologies. The key idea of the proposed technique is to ascertain a similarity between two concepts of the input ontologies, which is based on their locality in the ontology structures. The site of a node, that represents a concept within an ontology structure, determines its neighbour concepts. The meaning of the concept is also characterized by a linguistic analysis of the concept with respect to a large-scale dictionary like WordNet, to a corpus of documents, to manual rules, to lexical distances, etc. The proposed technique accepts the sources of background knowledge in order to establish a similarity measure. Graph matching techniques was used in order to examine the similarity of the location of two input concepts. In this exertion we present a new ontology mapping technique which, given two input ontologies, is able to map concepts in one ontology onto those in the other, without any user intrusion. It is based on association rule mining applied to the concept hierarchies of the input ontologies. We also present investigational results that demonstrate the accuracy of the proposed technique.

Keywords :— Ontology Mapping, Interoperability, Association Rule Mining.

I. INTRODUCTION

Data mining is an interdisciplinary subfield of computer science. It is the computational process of discovering patterns in large data sets involving methods at the meeting point of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, mode land inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Keyword technology (integrated with a series of statistical elements such as PageRank has the enormous advantage of being straightforward, easily applicable to many languages and very fast. When applied to the web, keyword technology took advantage of the free and voluntary labor hours of hundreds of millions of people. People, who by searching and clicking on one or more results, provide creators with an enormous quantity of information every day, This kind of information is priceless and helps to re-organize search results in the best possible way. Most knowledge on the Web is encoded as natural language text, which is convenient for human users but very difficult for software agents to understand. Even with increased use of XML-encoded information, software agents still need to process the tags and literal symbols using application dependent semantics. The Semantic Web offers an approach in which knowledge can be published by and shared among agents using symbols with a well defined, machine-interpretable semantics. At the core, a semantic search engine has the ability to understand the relationships between keywords, phrases or parts of speech within a search phrase, therefore allowing it understand the underlying meaning of the entire phrase. For example, a semantic search engine would be able to easily distinguish the differences between the following phrases made up of the same keywords but with obvious different implications.

The Semantic Web aims to achieve better data automation, reuse and interoperability. The main advantage of Semantic Web is to enhance search mechanisms with the use of Ontology's. Ontology is a general description of all concepts as well as their relationship. The Resource Description Framework /Schema (RDF(S)) and Web Ontology Language (OWL) are W3C recommended data representation models which are used to represent the ontology's. The basic method for constructing the Semantic Web is to use the terms defined in ontology as metadata to markup the Web's content. It is generally accepted that ontology refers to a formal specification of conceptualization Ontologies have been shown to be beneficial for representing domain knowledge, and are quickly becoming the backbone of the Semantic Web. Building ontologies, however, represents a considerable challenge for a number of reasons. It takes a considerable amount of time and

II. METHODOLOGY

2.1 ONTOLOGIES FOR THE SEMANTIC WEB

Conceptual structures that define an underlying ontology are germane to the idea of machine processable data on the Semantic Web. Ontologies are (meta)data schemas, providing a controlled vocabulary of concepts, each with an explicitly defined and machine processable semantics. By defining shared and common domain theories, ontologies help both people and machines to communicate concisely supporting the exchange of semantics and not only syntax. Hence, the cheap and fast construction of domain-specific ontologies is crucial for the success and the proliferation of the Semantic Web.

Though ontology engineering tools have become mature over the last decade, the manual acquisition of ontologies still remains a tedious, cumbersome task resulting easily in a knowledge acquisition bottleneck. Having developed our ontology engineering workbench, OntoEdit, we had to face exactly this issue, in particular we were given questions like

- Can you develop an ontology fast? (time)
- Is it difficult to build an ontology? (difficulty)
- How do you know that you've got the ontology right? (confidence)

In fact, these problems on time, difficulty and confidence that we ended up with were similar to what knowledge engineers had dealt with over the last two decades when they elaborated on methodologies for knowledge acquisition or workbenches for defining knowledge bases. A method that proved extremely beneficial for the knowledge acquisition task was the integration of knowledge acquisition with machine learning techniques [12]. The drawback of these approaches, e.g. the work described in [6], however, was their rather strong focus on structured knowledge or data bases, from which they induced their rules.

In contrast, in the Web environment that we encounter when building Web ontologies, the structured knowledge or data base is rather the exception than the norm. Hence, intelligent means for an ontology engineer takes on a different meaning than the — very seminal — integration architectures for more conventional knowledge acquisition [1].

Our notion of Ontology Learning aims at the integration of a multitude of disciplines in order to facilitate the construction of ontologies, in particular machine learning. Because the fully automatic acquisition of knowledge by machines remains in the distant future, we consider the process of ontology learning as semi-automatic with human intervention, adopting the paradigm of balanced cooperative modeling [5] for the construction of ontologies for the Semantic Web. This objective in mind, we have built an architecture that combines knowledge acquisition with machine learning, feeding on the resources that we nowadays find on the syntactic Web, viz. free text, semi-structured text, schema definitions (DTDs), etc. Thereby, modules in our framework serve different steps in the engineering cycle, which here consists of the following five steps (cf. Figure 1):

First, existing ontologies are imported and reused by merging existing structures or defining mapping rules between existing structures and the ontology to be established. For instance, [9] describe how ontological structures contained in Cyc are used in order to facilitate the construction of a domain-specific ontology. Second, in the ontology extraction phase major parts of the target ontology are modeled with learning support feeding from web documents. Third, this rough outline of the target ontology needs to be pruned in order to better adjust the ontology to its prime purpose. Fourth, ontology refinement profits from the given domain ontology, but completes the ontology at a fine granularity (also in contrast to extraction). Fifth, the prime target application serves as a measure for validating the resulting ontology [11]. Finally, one may revolve again in this cycle, e.g. for including new domains into the constructed ontology or for maintaining and updating its scope.

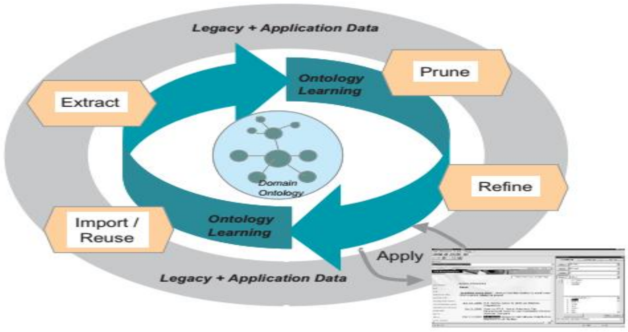

Figure2.1. 1: Ontology Learning process steps

effort to construct ontology, and it necessitates a sophisticated understanding of the subject domain. Thus it is an even greater challenge if the ontology engineer is not familiar with the domain. However, one of the major advantages claimed of ontologies is the potential for the "reuse" of knowledge.

2.2 AN ARCHITECTURE FOR ONTOLOGY LEARNING

Given the task of constructing and maintaining an ontology for a Semantic Web application, e.g. for an ontology-based knowledge portal that we have been dealing with [10], we have produced a wish list of what kind of support we would fancy.

Ontology Engineering Workbench OntoEdit. As core to our approach we have built a graphical user interface to support the ontology engineering process manually performed by the ontology engineer. Here, we offer sophisticated graphical means for manual modeling and refining the final ontology. Different views are offered to the user targeting the epistemological level rather than a particular representation language. However, the ontological structures built there may be exported to standard Semantic Web representation languages, such as OIL and DAML-ONT, as well as our own F-Logic based extensions of RDF(S). In addition, executable representations for constraint checking and application debugging can be generated and then accessed via SilRi1, our F-Logic inference engine, that is directly connected with OntoEdit.

The sophisticated ontology engineering tools we knew, e.g. the Proteg´ e´ modeling environment for knowledge-based systems [2], would offer capabilities roughly comparable to OntoEdit. However, given the task of constructing a knowledge portal, we found that there was this large conceptual bridge between the ontology engineering tool and the input (often legacy data), such as Web documents, Web document schemata, databases on the Web, and Web ontologies, which ultimately determined the target ontology. Into this void we have positioned new components of our ontology learning architecture (cf. Figure 2). The new components support the ontology engineer in importing existing ontology primitives, extracting new ones, pruning given ones, or refining with additional ontology primitives.

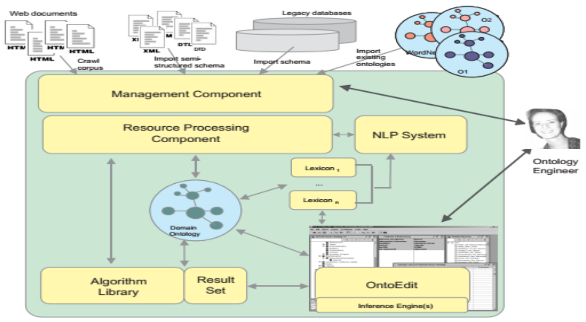

Figure 2.2.1: Architecture for Learning Ontologies for the Semantic Web

This structure corresponds closely to RDFS, the one exception is the explicit consideration of lexical entries. The separation of concept reference and concept denotation, which may be easily expressed in RDF, allows providing very domain-specific ontologies without incurring an instantaneous conflict when merging ontologies — a standard request in the Semantic Web. For instance, the lexical entry “school” in one ontology may refer to a building in ontology A, but to an organization in ontology B, or to both in ontology C. Also in ontology A the concept refered to in English by “school” and “school building” may be referred to in German by “Schule” and “Schulgeb¨aude”. Ontology learning relies on ontology structures given along these lines and on input data as described above in order to propose new knowledge about reasonably interesting concepts, relations, lexical entries, or about links between these entities — proposing the addition, the deletion, or the merging of some of them. The results of the ontology learning process are presented to the ontology engineer by the graphical result set representation. The ontology engineer may then browse the results and decide to follow, delete, or modify the proposals in accordance to the purpose of her task.

2.3 COMPONENTS FOR LEARNING ONTOLOGIES

Integrating the considerations from above into a coherent generic architecture for extracting and maintaining ontologies from data on the Web we have identified several core components. There are,

(i), a generic management component dealing with delegation of tasks and constituting the infrastructure backbone,

(ii), a resource processing component working on input data from the Web including, in particular, a natural language processing system,

(iii), an algorithm library working on the output of the resource processing component as well as the ontology structures sketched above and returning result sets also mentioned above and,

(iv), the graphical user interface for ontology engineering, OntoEdit.

Management component. The ontology engineer uses the management component to select input data, i.e. relevant resources such as HTML & XML documents, document type definitions, databases, or existing ontologies that are exploited in the further discovery process. Secondly, using the management component, the ontology engineer also chooses among a set of resource processing methods available at the resource processing component and among a set of algorithms available in the algorithm library.

Furthermore, the management component even supports the ontology engineer in discovering task-relevant legacy data, e.g. an ontology-based crawler gathers HTML documents that are relevant to a given core ontology and an RDF crawler follows URIs (i.e., unique identifiers in XML/RDF) that are also URLs in order to cover parts of the so far tiny, but growing Semantic Web.

Resource processing component. Resource processing strategies differ depending on the type of input data made available:

- HTML documents may be indexed and reduced to free text.
- Semi-structured documents, like dictionaries, may be transformed into a predefined relational structure.
- Semi-structured and structured schema data (like DTD's, structured database schemata, and existing ontologies) are handeled following different strategies for import as described later in this work.
- For processing free natural text our system accesses the natural language processing system SMES (Saarbr¨ucken Message Extraction System), a shallow text processor for German [7]. SMES comprises a tokenizer based on regular expressions, a lexical analysis component including various word lexicons, a morphological analysis module, a named entity recognizer, a part-of-speech tagger and a chunk parser.

After first preprocessing according to one of these or similar strategies, the resource processing module transforms the data into an algorithm-specific relational representation.

2.4 IMPORT & REUSE

Given our experiences in medicine, telecommunication, and insurance, we expect that for almost any commercially significant domain there are some kind of domain conceptualizations available. Thus, we need mechanisms and strategies to import & reuse domain conceptualizations from existing (schema) structures. Thereby, the conceptualizations may be recovered, e.g., from legacy database schemata, document-type definitions (DTDs), or from existing ontologies that conceptualize some relevant part of the target ontology.

In the first part of the import & reuse step, the schema structures are identified and their general content need to be discussed with domain experts. Each of these knowledge sources must be imported separately. Import may be performed manually — which may include the manual definition of transformation rules. Alternatively, reverse engineering tools, such as exist for recovering extended entity-relationship diagrams from the SQL description of a given database, may facilitate the recovery of conceptual structures.

In the second part of the import & reuse step, imported conceptual structures need to be merged or aligned in order to constitute a single common ground from which to take-off into the subsequent ontology learning phases of extracting, pruning and refining. While the general research issue concerning merging and aligning is still an open problem, recent proposals (e.g., [8]) have shown how to improve the manual process of merging/aligning. Existing methods for merging/aligning mostly rely on matching heuristics for proposing the merge of concepts and similar knowledge-base operations. Our current research also integrates mechanisms that use a application data oriented, bottom-up approach. For instance, formal concept analysis allows to discover patterns between application data on the one hand and the usage of concepts and relations and the semantics given by their heterarchies on the other hand in a formally concise way.

Overall, the import and reuse step in ontology learning seems to be the one that is the hardest to generalize. The task may remind vaguely of the general problems with data warehousing adding, however, challenging problems of its own.

2.5 EXTRACTING ONTOLOGIES

In the ontology extraction phase of the ontology learning process, major parts, i.e. the complete ontology or large chunks reflecting a new subdomain of the ontology, are modeled with learning support exploiting various types of (Web) sources. Thereby, ontology learning techniques partially rely on given ontology parts. Thus, we here encounter an iterative model where previous revisions through the ontology learning cycle may propel subsequent ones and more sophisticated algorithms may work on structures proposed by more straightforward ones before.

Describing this phase, we sketch some of the techniques and algorithms that have been embedded in our framework and implemented in our ontology learning environment Text-To-Onto. Doing so, we cover a very substantial part of the overall ontology learning task in the extraction phase. Text-To-Onto proposes many different ontology components, which we have described above (i.e. L; C; R;:::), to the ontology engineer feeding on several types of input.

III. CONCLUSION

On the Semantic Web, data has structure and ontologies describe the semantics of the data. The main aim of this work was that using these semantic metaphors of the data, more efficient discovery and mining of resources is possible. We introduced an ontology-based crawler for the Semantic Web. A lot of work have already been done on the discovery of resources on the Web using different sorts of crawlers. Furthermore, we exploit the semantic data and structure of the Semantic Web to determine and extract resources more efficiently. We have opted for a focused crawler as we want to discover and extract information during the crawl. Semantic matching between downloaded web page contents and ontologies guides the crawler for extracting applicable information which provide scope for better search engine.

IV. FUTURE WORKS

How to mine the features reasonably will be investigated in the future work. There are a number of future research guidelines to extend and improve to this work. One direction that this work might continue on is to improve on the accuracy of similarity calculation between documents by employing dissimilar similarity calculation strategy. Although the current system proved more accurate than habitual methods, there are still rooms for enhancement.

V. REFERENCES

[1] B. Gaines And M. Shaw. Integrated Knowledge Acquisition Architectures. Journal Of Intelligent Information Systems, 1(1), Pages 9-34.

[2] E. Grosso, H. Eriksson, R. Fergerson, S. Tu And M. Musen. Knowledge Modeling At The Millennium — The Design And Evolution Of Prot´Eg´E-2000. In Proceedings Of Kaw-99, Banff, Canada,.

[3] U. Hahn And M. Romacker Content Management In The Syndikate System — How Technical Documents Are Automatically Transformed To Text Knowledge Bases. Data & Knowledge Engineering, 35, Pages 137-159.

[4] G. Miller. Wordnet: A Lexical Database For English. Cacm, 38(11), Pages 39-41.

[5] S.Sudha, Lakshmi.S, "Comparison Of Ontology Ranking Algorithms On Semantic Web" International Journal Of Advanced Research In Computer And Communication Engineering,Vol. 5,Issue 4,April 2016.

[6] Guangfei Yang, Kaoru Shimada, Shingo Mabu And Kotaro Hirasawa," A Personalized Association Rule Ranking Method Based On Semantic Similarity And Evolutionary Computation" , Ieee Conference,.2010

[7] Harith Alani ,"Ontology Construction From Online Ontologies", In Proc. 15th International World Wide Web Conference,2006
.

[8] Harith Alani, Christoper Brewster And Nigel Shadbol,"Ranking Ontologies With Aktive Rank",2007.

[9] Huang Xiao-Xi, Zhou Chang-Le ,"An Owl-Based Wordnet Lexical Ontology" Journal Of Zhejiang University Science ,2007.

[10] Iván Cantador, Miriam Fernández, Pablo Castells, "Improving Ontology Recommendation And Reuse In Web Core By Collaborative Assessments", 2007 Symposium On Intelligent Information Technology Application, 2009