

OntoGenie : Extracting Ontology Instances from WWW

Chintan Patel, Kaustubh Supekar, and Yugyung Lee

School of Interdisciplinary Computing and Engineering
University of Missouri-Kansas City
{copdk4, kss2r6, leeju}@umkc.edu

Abstract. Web has become a wildly huge information source on planet. The problem being that the information is not machine perishable. Standardized Ontological representation of knowledge solves the problem as proposed by Semantic Web. The major challenge remaining is to export information present on current Web as Ontological data for Semantic Web. We have developed a solution, OntoGenie, that scans the Web pages to create knowledge instances for given Ontologies. We used WordNet as the mapping bridge between the Ontologies and the Web page terms. OntoGenie was tested over currently available Ontologies and some interesting and motivating results were obtained.

1 Introduction

The potential of Semantically enriched Web is beyond imagination, the automation that can be achieved by Semantic Web technologies is astonishing, ranging from simple semantic searches to automatically discoverable and invocable Semantic Web services. The knowledge in Semantic Web is encoded in webized way, as simple directed graphs. The semantics are attributed to the graph by the conferring standardized meanings to nodes and arcs. Resource Description Format (RDF) provides the basic infrastructure for representing the meta-data over Web [1]. To add more semantics and vocabulary for the RDF data, layers of Web Ontology (Web Ontology Language) and RDF Schema (RDFS) language respectively were added on Semantic Web stack [2]. Ontologies are the key idea that enables conceptualization and representation in a given domain. Ontologies provide the explicit formalization and specification of the concepts and their corresponding relationships. However it should be noted that Ontologies have associated specific instantiation for the concepts defined. Moreover, most of the knowledge queries are over instances of the Ontological conceptualization. Consider the following query:

Query: List all the universities located in Boston ?

Here, the query requires to interpret the instance, Boston, as a City and retrieve all instances of Universities located in the City (e.g. MIT, Harvard, etc). It can be conjectured that future Semantic Web will contain few domain Ontologies and a large number of instance data. For e.g. the domain of University

can contain over 1000 instances. Moreover, Ontologies are largely developed manually by Domain Expert, filling in the instance data manually is an arduous task. One cannot expect a sane human being to go around finding each and every university and instantiating the relevant concepts.

We believe that process of creating Ontology instances can be automated from data extracted from natural language pages on World Wide Web (WWW). Also, to accelerate the nurturing and growth of Semantic Web, there is a pressing need to develop tools that would provide smooth transition from current Web to Semantic Web.

We have developed a tool, OntoGenie, that uses WordNet¹ to convert *data extracted from Web* to *structured knowledge*. The tool was developed as a part of ongoing BEE-SMART (A Natural Language Interface to Semantic Web) project at University of Missouri². In this paper we describe the architecture of the tool and the results obtained.

2 OntoGenie Functionality: What's your wish master?

The OntoGenie is a semi-automatic tool that takes as input the Ontology and Web pages (or plain natural language texts) and generates Ontology Instances (OI) as output. The tool uses WordNet as the mapping bridge between Ontologies and Web data. The overall OntoGenie functionality is elaborated in the following algorithm.

2.1 OntoGenie Algorithm (Descriptive)

Step 1: [Map the concepts in Ontology into WordNet taxonomy] Retrieve the Concepts CO_i from Input Ontology, O and map it into a concept CW_i in WordNet taxonomy. Retrieve the Concepts CO_i from Input Ontology, O and map it to a concept CW_i in WordNet taxonomy.

- The mapping is initially performed by first canonizing the English terms defining the Concepts (CO and CW).
- Usual case is that many terms in WordNet map into same concept. For e.g. the concept has more than one senses in WordNet, it can mean an "educational institution" or a "group of persons associated by some common tie".

In short following mappings are generated in this step

$$CO_i \rightarrow CW_j \quad (1)$$

where i, j refers to i_{th} and j_{th} concepts in input Ontology and WordNet taxonomy respectively.

Step 2: [Capture the terms occurring in Web pages]

¹ <http://www.cogsci.princeton.edu/wn/>

² <http://sice527.ddns.umkc.edu/BeeSmart/>

- Web pages for the domain are extracted using some existing search service, OntoGenie interfaces the Google Web service³ to retrieve the Web pages pertaining to a particular domain. As a plugin we also used dmoz directory⁴ to retrieve Web pages for a particular domain.
- The web pages are scanned word by word, each w_i (Web page word) is canonized and compared with the hyponyms of the CW_i present in the WordNet. Hyponyms are the word that are more specific than the given word, essentially closer to the instance space.

$$w_i - > Hyponym(CW_i) \quad (2)$$

Where w_i is the word in the Web page being mapped to the hyponym of the WordNet concept. We basically assume the transitive relationship to hold among the Ontology Concepts (CO) and WordNet Concepts (CW) and Web page terms. For e.g. Concept Country in the Ontology (fig x) is mapped on to similar concept, Country in WordNet, CWi. Later, USA being discovered as Country from the WordNet taxonomy, we create USA as instance for concept of Country i.e. we now have the transitive mapping :

$$w_i - > O_i \quad (3)$$

that maps a natural language terminology to Ontology concept. Also to be noted is that hyponym relationships is also a transitive relation being encountered in the WordNet taxonomy

Step 3: [Discovering relationships]

- a. Once the mappings are accomplished for a page, we discover the relationships that holds between the instance of the concepts extracted. Conventionally, the task of discovering relationships was done via morphologically determining the verbs and the relationships to noun[[3], [?]]. The approach works for simple "toy" cases, but fails practically in real world cases, with large amount of valid OIs going undetected.
- b. We propose to use a simple approach using principle of locality, the idea is to blindly assume set of concepts discovered in predetermined locus around the concepts to be related. To better understand the idea, consider the Ontology concepts being a graph, with Concepts represented as nodes and the Relationships as links. Distance between set of Concepts can be defined as number of links encountered traversing between the Concepts (we assume the shortest path).

$$\delta = \text{number of links between}(CO_i \text{ and } CO_j)$$

Now we put a predetermined "locality" constraint saying that our domain of interest for a discovered concept, CO_k , will consist of all the concepts at a distance. So for e.g. as described in [?] , we discovered an instance of University (MIT) and an instance of Country, we can assume a relationship

³ <http://www.google.com/apis/>

⁴ <http://dmoz.org/>

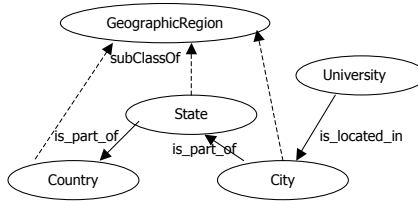


Fig. 1. University Ontology Excerpt

to hold between them. It should be noted however that if we don't find the instances of intermediate nodes (for e.g. State in this case) we still consider them as blank nodes. Such blank nodes can be filled on while scanning other Web pages for the given domain. The purpose of incorporating the principle of locality was to make sure that we don't end up discovering largely disconnected knowledge instances

3 OntoGenie Implementation : Rubbing the Lamp !

OntoGenie architecture has been designed to exploit the functionality provided by the existing available tools. The WordNet was interfaced in Java, and Jena was used for Ontology manipulation. With Jena we perform following functionalities :

- a. Parse and extract concepts in Ontologies (DAML, RDFS)
- b. Discovering distances (δ), *basically finding the distance between the Concepts and traversing by properties*
- c. Creating and Validating the Ontology Instances (RDF)

To disambiguate the Concept mappings to WordNet, as mentioned in Step 1. we have developed a intuitive interface (fig x) for a domain expert to select the right sense for the automatically discovered mappings. We used KAON[?] as our backend data store (mandated by the requirements of the overall project). To interface Google Web service, we used Java Web Services Developer Pack⁵ (JWSDP).

⁵ <http://java.sun.com/webservices/webservicespack.html>

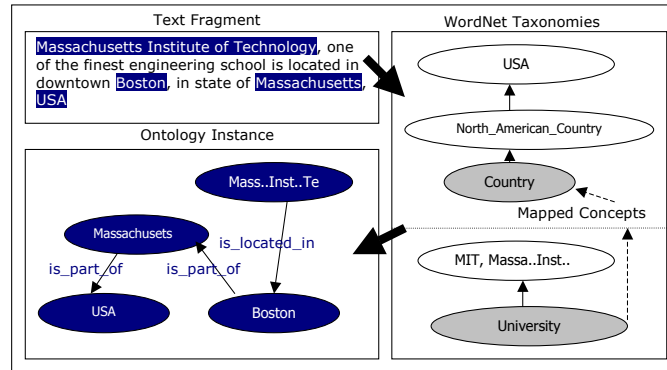


Fig. 2. Flow of OntoGenie Algorithm

4 OntoGenie Testing and Results : Your wish is my Command !

The OntoGenie is being successfully used to discover Knowledge Instances from the Web. We discovered some interesting chunks of knowledge from web pages. We tested the framework with University Ontology⁶ and extracted the university related web pages from [?] .

Figure x, depicts one of the RDF instance being scooped from the Web pages

The major drawback of the current OntoGenie algorithm being, that it doesn't do any kind of checking or semantic checking for verifying relationships among two concepts, for e.g. if we have Ontology saying

.University.has_dept.Mathematics

As of now, OntoGenie will identify "Geometry" as instance of Mathematics and this leads to inconsistency. We are currently exploring techniques to disambiguate the correct instantiation. Another notable problem is related to the performance issue of mapping (and canonizing) the Web page words to match the Ontology-WordNet mappings, the time complexity runs very high. The reason being we currently examine each and every word (also set of words) in the Web page to perform the matching task.

5 OntoGenie : Conclusions : Getting back into Lamp !

We presented a simple, practical and implemented framework, OntoGenie that solves the highly critical and important problem of discovering Knowledge in-

⁶ <http://www.cs.umd.edu/projects/plus/DAML/onts/cs1.0.daml>

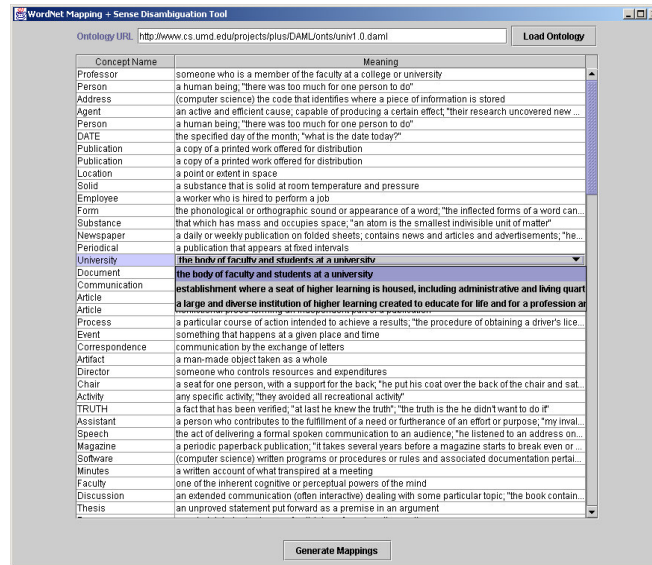


Fig. 3. Disambiguating Ontology to WordNet mappings

stances from Web. OntoGenie, algorithm is based on transitive mapping from Ontology to Web page terms using WordNet as a effective bridge. We gave implementation details and a glimpse of the results obtained. Currently we are exploring set of other tools (GATE⁷) to assist in resolving some of the unresolved issues.

References

1. RDF, Resource Definition Framework, <http://www.w3.org/RDF/>
2. Tim Berners Lee, Semantic Web Roadmap <http://www.w3.org/DesignIssues/Semantic.html>
3. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery, Learning to Extract Symbolic Knowledge from the World Wide Web, Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98).
4. Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel R. Shadbolt, Automatic Ontology- Based Knowledge Extraction from Web Documents, pg 14-21 IEEE Intelligent, Jan/Feb 2003
5. Raphael Volz, Daniel Oberle, Steffen Staab, Boris Motik, KAON SERVER - A Semantic Web Management System, WWW2003, May 20-24, 2003, Budapest, Hungary.
6. College and Universities Listings, <http://www.mit.edu:8001/people/cdemello/univ-full.html>

⁷ <http://gate.ac.uk/>