RIT Department Of Computer Science
MSc Project Pre-Proposal:
P-DeepMiner – Scaling up DeepMiner using map reduce to handle large number of pages

Varun Goyal
May 14, 2012
Chair: Dr. Xumin Liu

Problem:

The vision of Semantic Web services promises a network of interoperable Web services over different sources. A major challenge to the realization of this vision is the lack of automated means of acquiring domain ontologies necessary for annotating web pages. Wensheng Wu et al. proposes DeepMiner [1], which learns domain ontologies from the source Web sites to discover domain concepts and instances and thus develop the domain ontology. We propose P-DeepMiner to effectively learn domain ontology from interfaces and data pages of a set of domain sources. Achieving this goal requires P-DeepMiner to make several innovations.

- Ontology Learning: The knowledge acquired from source interfaces only is often incomplete since data pages of the sources may contain additional information. Further, different sources may contain a different set of concepts and instances. As such, P-DeepMiner will learn the base domain ontology using nearest neighbor learning (kNN) algorithm [2], then it will grow the current ontology by probing the sources and learning additional concepts and instances from the data pages retrieved from the sources.

- Handling heterogeneities among sources: Due to the autonomous nature of sources, the same concept may be represented quite differently over different sources. Another major challenge is thus to identify the semantic correspondences between concepts learned from different sources. To address this challenge, P-DeepMiner will use WordNet [3] to effectively discover unique concepts over different interfaces. The learned ontology is then exploited for discovering new concepts and instances from data pages.

This approach can be scaled to a large number of Web sites for P-DeepMiner to be used effectively and thus produce efficient ontologies. This can be done using mapreduce technique.

There have been some efforts in learning domain ontology for Web services. Our work is most closely related to DeepMiner [1], but different in several aspects. First, [1] learns the base ontology in a snowballing fashion however we are using kNN algorithm for faster processing. Second, we use WordNet to discover unique concepts over different interfaces as WordNet provides us with a large lexical database of English grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept thus making it easier and faster to discover the concepts.

[5] Employs several machine learning algorithms for the semantic annotation of attributes in source interfaces. The annotation relies on manually constructed domain ontology. Our work is complementary to these works in that we aim to automatically learn domain ontology from the

information on the source Web sites. The learned ontology can then be utilized to annotate the Web services.

Methodology

MapReduce is a framework for processing such problems across huge datasets using a large number of computers (nodes) [6], collectively referred to as a cluster. If all nodes are on the same local network and use similar hardware, it is known as a cluster. Basic steps in implementing mapreduce on DeepMiner would be:

> **Map step:** Given a Set of domain the master node will take parsed DOM tree of the HTML web page, and learn the base ontology O using kNN algorithm. Now this base ontology along with individual source pages is distributed to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node will then process these web pages to build the sub-ontologies, using the base ontology and probing through the web-page and its instance to develop concepts with the help of WordNet. These sub-ontologies are passed back to the master node.
>
> **Reduce step:** The master node will then collect all these sub-ontologies and combine them with the help of WordNet to form the final ontology for the website.

Mapreduce will allow for distributed processing of each webpage using the map and reduction operations. Each web page will have its own set of attributes and concepts making it independent of other to form sub-ontology. Once the sub-ontology is created in the map stage, the new-found concepts can be merged considering their semantics in the reduce step.

Evaluation:

We will look for websites, which contain a huge data set for various domains like amazon.com, ebay.com for e-commerce domain. The evaluation for the resulting tool would be two fold:

1. Performance:
    a. *Precision*: which is the percentage of correct mappings of attributes among all the mappings identified by the system, and recall, which is the percentage of correct mappings among all mappings given by domain experts.
    b. *Identifying data regions*: we randomly select five sources for each domain. For each source, query submission is made by automatically formulating a query string (consists of form element names and values) then posing the query to the source. If an attribute does not have instances in its interface, the instances of its similar attributes (available from the base ontology) are used instead. This probing process is repeated until at least one valid data page is returned from the source, judged based on several heuristics as employed in [7].
    c. *Discovering concepts and their instances from data pages*: This is done by first manually determining the number of concept labels and their instances in all data pages, and then comparing the concept-instances pairs discovered by P-DeepMiner.

2. Efficiency: To check for improvement in the over-all efficiency we will run both DeepMiner and P-DeepMiner and note the time taken for both tools to create an ontology, and thus compare their run time on various attributes like single processor and 8 processor cluster.

References:

1. W. Wu, A. Doan, C. Yu, and W. Meng. Bootstrapping domain ontology for Semantic Web services from source web sites. In Proceedings of the VLDB-05 Workshop on Technologies for E-Services, 2005.

2. Yi-Ching Liaw, Maw-Lin Leou, Chien-Min Wu, Fast exact k nearest neighbors search using an orthogonal search tree, Pattern Recognition, Volume 43, Issue 6, June 2010, Pages 2351-2358, ISSN 0031-3203, 10.1016/j.patcog.2010.01.003. (http://www.sciencedirect.com/science/article/pii/S0031320310000336)

3. WordNet by Princeton University (http://wordnet.princeton.edu/)

4. A. Patil, S. Oundhakar, A. Sheth, and K. Verma. METEOR-S: Web service anno- tation framework. In WWW, 2004.

5. A. Heß and N. Kushmerick. Machine learning for annotating semantic web services. In AAAI Spring Symposium on Semantic Web Services, 2004.

6. Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (January 2008), 107-113. DOI=10.1145/1327452.1327492 (http://doi.acm.org/10.1145/1327452.1327492)

7. S. Raghavan and H. Garcia-Molina. Crawling the hidden Web. In VLDB, 2001.