Annotation of Hyperlinks with Semantic Meaning

Bilal Gonen[1], **Samir Tartir**[2], **Ravi Pavagada**[3]
[1]Computer Science and Engineering Department, University of Nevada, Reno, Reno, Nevada, U.S.A.
[2]Department of Software Engineering, Philadelphia University, Amman, Jordan
[3]IT, Excelacom Inc, Reston, VA, U.S.A.

Abstract— *Web pages in the web represent certain concepts in the domain they fall in, and the connections between them represent the relations between the concepts they represent. In the current web, people are using links blindly without knowing what these links point to, or what kind of relationship this link represents. With the advent of the semantic web, concepts and relationships among them are represented in an ontology. This can be utilized to make links more meaningful. Web pages can be searched, browsed or even reorganized based on their concept and relationship labels. Links in a webpage can render useful information about the page it is pointing to. We can annotate a webpage and its links with appropriate concepts from ontology. This paper presents a new idea of propagating concept from a webpage to the links pointing to that page or from the links to the webpage. Propagation of concepts is based on certain criteria which will be discussed later in this paper. We also propose a new idea of automated voting which is used to choose the right concept or relation from a number of concepts and relation matches.*

Keywords: semantic web, link annotation, data mining

1. Introduction

The network of hyperlinked documents, as it exists now, lacks semantic information in machine understandable form. It can only be browsed or searched by keywords -not concepts. There exist projects that automatically or semi-automatically annotate web pages with concepts taken from ontology. This effort makes web pages more understandable for machine processing and searching. In our project we would like to focus more on navigational implications of adding semantic annotation to web pages. Currently user or machine navigates between web pages by traversing them via hyperlinks. Decision if accessed page is relevant to the undertaken search can be made only after retrieving and analyzing the destination web page. In our project, we would like to add more semantic meaning to links themselves on the source page, so concepts included on target page can be evaluated without retrieving page itself.

In this paper, we use well formed computer science department ontology to annotate links and web pages with concepts. Web pages and links of the page can then be associated with concepts and relations from ontology. For example, web pages from computer science department of University of Georgia web site can be associated with concepts such as faculty, department, course, lecturer, research assistant etc... These web pages can therefore be treated as concept instances. Relationship can be defined between a webpage and its link. For instance, a student's webpage might have a link to his course page. In ontology there could be a relation say "takes" between the student and the course. This information will be annotated in the links along with the link concept "course". We have used ontology dictionary which associates labels to each concept and relations in the ontology. These labels are very useful in concept matching. Labels play the key role, since they are matched with the page contents and link window to extract appropriate concepts. We havenŠt used NLP techniques to get the concept matches. Our goal is to start with set of plain, connected web pages and by extracting information and matching them with the ontological concepts and also annotate the links with concepts and relations joining them. In this project, we would like to utilize already known algorithms and solution for page annotations. We think that combining different approaches of page annotation and information/concept propagation between web pages can improve the overall quality of annotated data.

Paper is organized as follows. In Section 2, we describe the related work and our ideas. Section 3 briefs our work and discusses the architecture of the proposed system. Section 4 explains the approach we took in building the proposed system. Section 5 describes propagation and voting schemes used. Section 6 describes the testing and experimental results, and Section 7, Conclusion and Future work.

2. Related Work and Our Ideas

There are papers on HTML Tag tree extraction or deriving link context [1] [2] [3]. One of them is ŞDeriving link-context from HTML tag treeŤ by Gautam Pant et al [4]. There are also other papers on automated semantic annotations [5] [6] [7] [8] [9] [10]. "SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotations" [11] talks about automation of web page annotations. "Mining the link structure of semantic web", by Souman Chakrabarti et al. [12] talks about HITS algorithm which takes advantage of the hubs in some fields and uses techniques that take advantage of social organizations of the web and allocates weights for the hub pages and

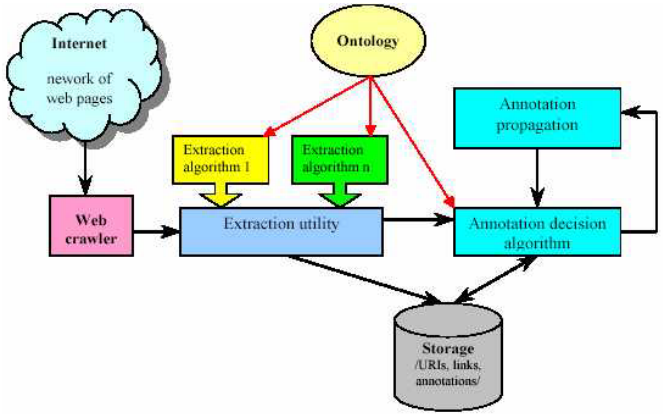

Fig. 1: General System Architecture

authorities in iterative process. The paper "On extracting link information by relationships instances from a website" by Myo-Myo Naing et al [13] talks about a web page which is being associated with a concept in ontology and links two different web pages based on the relationship between concepts in the ontology.

Our work is slightly different from their work. We incorporate voting of relations whenever there are more than one relation matches between two concepts. Concept matching is an area in itself and there are lots of papers on it. There are lots of AI and natural language processing techniques used to achieve this. As mentioned previously, we have concentrated more on concept labels defined in the ontology to find a concept match. Our work concentrate more on the information which is around the link, i.e. link context and match the link to a concept. New idea of concept propagation is proposed which would propagate concepts from a Webpage to the links pointing to that page if there is a tie in the number of concept matches for the given set of links. Propagation from links to page is done if most of the links agree on a single concept. Voting of concepts and relations is done whenever there is ambiguity.

3. Architecture Overview at a high level

We would like to make our system modular and expandable for future needs. As we cannot modify the content of web pages, we can only keep discovered annotations of pages and links in snapshot of selected web pages.

3.1 WebCrawler

Web Crawler, crawls the web structure and supplies the raw data for further analysis. HTML from web pages is analyzed by extraction utility. The extraction mechanism tries to match the whole page to some concepts in ontology.

```
<rdfs:Class rdf:about="http://protege.stanford.edu/kb#AssistantProfessor"
rdfs:label="Assistant Professor">
<rdfs:label>Assistant Prof</rdfs:label>
<rdfs:label>Assistant Faculty</rdfs:label>
<rdfs:subClassOf rdf:resource="http://protege.stanford.edu/kb#professor" />
</rdfs:Class>
```

Fig. 2: Labels for a concept

3.2 Ontology Dictionary

Ontology dictionary is the key part of our project. Dictionary labels are assigned for concepts and relations in the ontology. Dictionary labels for relations and concepts are comprised of hypernyms, synonyms and homonyms. We have added RA as a label for Research Assistant, TA as a label for Teaching Assistant etc.

We add labels to each of the concept in the ontology. Figure 2 is an extract from the ontology which describes an rdf:class Assistant Professor and its associated labels namely Assistant Professor, Assistant Faculty etc.

3.3 Extraction utility

Extraction utility is comprised of page and link analyzer, which analyses the page for the tags and assembles a vector of number of concept matches for each tag. The vector size is determined by the number of concepts in the ontology. We have prioritized various html tags in the webpage based on its importance. For example. <Title>, <Head>, and <Body> tag are given the most importance. It also tries to categorize links in this web page based only on information contained in the link window. Link extractor extracts the text of information based on the window sizes or number of bytes of text before and after the link. It then assembles the concept weights based for each of the link window sizes namely 0, 50 (25 words before and after the link) and 150 (75 words before and after the link).

3.4 Annotation decision

Voting is done whenever there is more than one concept matching a given page or a link. Based on the values set in the configurator, i.e. relative importance of tags or it could be based on relative size of the windows 0, 50 or 150, the voter calculates the new vote by calculating the product of the weight vector to the weights assigned in the configurator. Our configurator is flexible and easier to change. We have assigned weights of 0.5, 0.3, and 0.2 weights for anchor text window sizes of 0, 50 and 150 respectively. We have assigned weights of 0.4, 0.3, and 0.3 for "title", "head", and "body" tags respectively.

3.5 Database Storage for persistence

Once the voting is done for the web pages and links, we update the webpage and links table in the database. All extracted information is stored in persistent storage along with the matched concept for web page and its links. One advantage of our approach is that we have designed the

project in such a manner that the all the web page and web link information are stored in the tables of our database. We crawl the web pages and load the tables in the database with the concepts. Then we follow the links to store its content and the relevant concept matches. Then propagation tables of the web pages and the links are updated along with the concept and their relation matches.

3.6 Propagation of concepts

Finally, the decision and propagation loop occurs. At this final step, the extracted information is analyzed again. Some of the extracted information may be deleted from page; some can be inferred or pushed from links to page. In this step, web pages are analyzed in network and we allow annotation flow between nodes. Both from page to describing link and from link to described page. This is an iterative process and in a few iteration the network reaches some stable (or near-to-stable) state. In such state, we say that the selected network is annotated and can be used in semantic navigation. Propagation of concepts and voting is discussed in section 5.

4. Approach

The approach will work in two general phases: Preparation and Annotation.

4.1 Preparation

Here a deep analysis of the Computer Science department in the University of Georgia will be conducted, resulting in building an ontology that represents the current structure of the department. This resulting ontology will be used in the next phase for annotation.

4.2 Annotation:

This is where the actual process of page and link annotation will take place. This phase is divided into three stages:

1) Page annotation
2) Link annotation
3) Relationship annotation

4.2.1 Page annotation

In this stage, all the pages in the Computer Science department site will be analyzed in one of the current methods, or a new method that we might need to develop. The result of this analysis will be a mapping between a certain page, and a node in the ontology designed in phase I.

4.2.2 Link annotation

Here, each page will be scanned for links that point to pages in the same domain, and each link will carry the annotation of the page it points to.

4.2.3 Relationship annotation

This is the final stage that defines which type of relationship the link defines. This relationship is obtained from the ontology based on the types (concepts) of the page with the link, and page the link points to. The resulted annotated pages will be stored in a database the application has access to write to and issue queries against.

5. Voting and Propagation

Voting for webpages, links and relationships are illustrated in algorithm 1, algorithm 2, and algorithm 3 respectively.

Algorithm 1 $Voting\ for\ Webpages$

1: **for all** tag entry in the configurator **do**
2: **for all** concept in the ontology **do**
3: Calculate the weights based on the number of matches
4: **end for**
5: **end for**
6: Select the maximum concept weight among all the vectors

Algorithm 2 $Voting\ for\ Web\ links$

1: **for all** window size of the hypertext **do**
2: **for all** concept in the ontology **do**
3: Calculate the weights based on the number of matches for each concept and weight assigned to each of the individual anchor text sizes.
4: **end for**
5: **end for**
6: Select the maximum concept weight among all the vectors

Algorithm 3 $Voting\ for\ Relations$

1: Given two concepts
2: Get all relations between the two concepts
3: Traverse all the concept nodes that are above a given two concept in the ontology and extract the relations between them
4: Match these relations with the text surrounding the hypertext window.
5: Choose the concept that has the maximum number of keyword matching among all the hypertext window vectors

The main drawback at this point with respect to relation voting is that we havenŠt concentrated on weighting relations between the concepts with different weights.

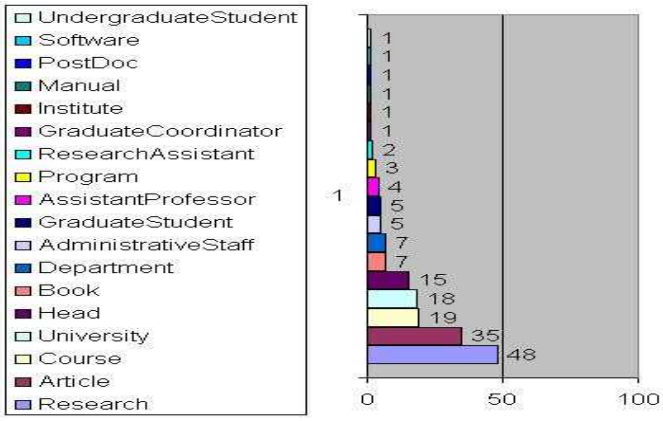

Fig. 3: Page concepts and number of matching pages

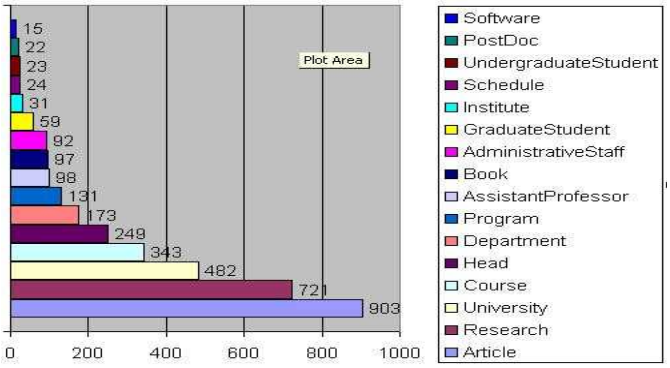

Fig. 4: Concepts and number of matching links

5.1 Propagation Algorithm

Propagation of concepts is done to increase the accuracy in the concept matches for a given web page and its links. Propagation is done after the voting stage. At the end of the voting we would have the concepts for web pages and its links stored in the database. Propagation of concepts are explained in algortihm 4:

Algorithm 4 *Propagation Algorithm*

1: Get the concept match to a page querying the database.
2: Query database and get the concept matching for the links pointing to a given page.
3: Propagate the concept from links to page if majority of links matches to a given concept.
4: If there is a tie, then propagate the concept from the webpage to the links. (Reason behind this is based on the intuition that webpage concept has higher priority than the concepts of the tied links.)

6. Testing and Experimental results

Testing was done on www.cs.uga.edu domain. Web crawler started crawling from http://lsdis.cs.uga.edu and was allowed to crawl only the CS domain. We ran our test crawling 200 web pages and the total number weblinks crawled were about 4599. We used the following test cases:

1) Finding concept for page
2) Finding concept for link
3) Propagating down concept, i.e. from a webpage to the the links pointing to it.
4) Propagating concept up, i.e. propogation occurs from the weblinks to the webpage.
5) Finding relation for a given link

Example for the best concept match.
URL: http://www.cs.uga.edu/academics/UGProgram/4900courses.htm

Fig. 5: ..

Total concept matches for the pages crawled were about 174. Concepts were not matched to a page since there was not any concept label matching to the pages.

There were about 3463 links matching to a concept out of a total of 4599.

Some of the good examples for concept matching before propagation;

- concept: “AssistantProfessor” for “http://webster.cs.uga.edu/ budak/”
- concept: Research” for “http://webster.cs.uga.edu/ Ekochut/Research”
- concept: Article” for “http://lsdis.cs.uga.edu/Projects/METEOR-S/Downloads”
- concept: Department” for “http://www.cs.uga.edu”

Propogation of Concepts

Some of these webpages initially had “Research” as the page

concept. After propogating from links to the page we had the following results. Below are some of the examples of propogating concepts from links to the pages.

- "Department" for "http://www.cs.uga.edu/ jam"
- "Department" for "http://lsdis.cs.uga.edu/ devp"
- "GraduateStudent" for "http://lsdis.cs.uga.edu/ mperry"
- "GraduateStudent" for "http://lsdis.cs.uga.edu aleman"
- "Department" for "http://lsdis.cs.uga.edu/ cthomas"
- "Department" for "http://lsdis.cs.uga.edu/ kunal"
- "Department" for "http://lsdis.cs.uga.edu/ kaarthik"
- "TeachingAssistant" for "http://lsdis.cs.uga.edu/ mperry"

For Relations

- The page "http://lsdis.cs.uga.edu/about/index.php?page=1" matched to a concept "AdministrativeStaff" and "http://www.uga.edu" matched to concept "University"
- There was only one relation between them in the ontology. The relation found between these two concepts were "works"

Summary of our experimental test cases: In our experiment, first of all, we crawled 200 web pages. We set starting pages as http://lsdis.cs.uga.edu and http://www.cs.uga.edu. We limited the crawling area as in the "cs.uga.edu" domain. After crawling 200 pages, we crawled the links within each pages. Thus, the total number of link crawled were 4599. By running our algorithm on these 200 webpages, our algorithm assigned some concepts to 174 pages of them. Also running our algorithm on the 4599 weblinks, our algorithm was able to assign concepts to 3463 links.

Propagation algorithm used showed good results. Below is an example of its effectiveness.
The Computer Science web page had concept "Research" before propagation. By applying the propagation method considering 93 pointing links, original page concept "Research" has been changed as concept "Department". Concept was propagated from Links to Page.

7. Conclusion and Future Work

Our approach of using ontology labels for relations and concepts in ontology was very beneficial in concept matching. We were able to match most of the web pages to the concept in the ontology. Labels were represented by hypernyms, synonyms and homonyms. Our propagation algorithm showed excellent results. We were able to compare the effectiveness of the algorithm by comparing it with concept before propagation. Voting was done based on the importance of individual tags and also based on the importance of the various anchor text window sizes. Relation voting seemed to work pretty well. Relation voting was done whenever there were more than one relation matches between two concepts. Our algorithm or methodology could be changed by adding different weights to relations between concepts, i.e. we traverse the ontology tree to find all the possible relations between the concepts by traversing the tree in a bottom up fashion. Our algorithm uses these relations and matches the keywords around the anchor text window. Our future work is to include different weights to the relations as we traverse the tree in a bottom up fashion. We still need to tune the ontology as there are no concept matches for some of the web pages crawled. Using various label names to a given concept may not be the best idea compared to NLP techniques. Our ontology is not populated with instances so that we could use it for semantic web search. Initial experimental results were very promising, and we wish to work on this further.

References

[1] G. Pant, S. Bradshaw, F. Menczer, Search engine-crawler symbiosis: Adapting to community interests, in: ECDL, 2003, pp. 221–232.

[2] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, J. Kleinberg, Automatic resource compilation by analyzing hyperlink structure and associated text, in: IN PROCEEDINGS OF THE SEVENTH INTERNATIONAL WORLD WIDE WEB CONFERENCE, 1998, pp. 65–74.

[3] I. Varlamis, M. Vazirgiannis, Web document searching using enhanced hyperlink semantics based on xml, in: Proceedings of the International Database Engineering & Applications Symposium, IDEAS '01, IEEE Computer Society, Washington, DC, USA, 2001, pp. 34–43.

[4] G. Pant, Deriving link-context from html tag tree, in: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, DMKD '03, ACM, New York, NY, USA, 2003, pp. 49–55. doi:http://doi.acm.org/10.1145/882082.882094.
URL http://doi.acm.org/10.1145/882082.882094

[5] L. Fraser, O. Fraser, C. Locatis, Effects of link annotations on search performance in layered and unlayered hierarchically organized information spaces, Journal of the American Society for Information Science and Technology 52.

[6] Semantic blogging and bibliography management, http://www.w3.org/.

[7] M.-M. Naing, E.-P. Lim, D. G. Hoe-Lian, Ontology-based web annotation framework for hyperlink structures, in: Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops) - (WISEw'02), WISEW '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 184–.
URL http://portal.acm.org/citation.cfm?id=832313

[8] P. Brusilovsky, J. Eklund, A study of user model based link annotation in educational hypermedia, Journal of Universal Computer Science 4 (1998) 429–448.

[9] M. Engelhardt, T. C. Schmidt, Semantic linking - a context-based approach to interactivity in hypermedia, Tech. Rep. cs.IR/0408001 (2004).

[10] S. Mukherjee, G. Yang, I. V. Ramakrishnan, Automatic annotation of content-rich html documents: Structural and semantic analysis, in: In Intl. Semantic Web Conf. (ISWC, 2003, pp. 533–549.

[11] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, , J. Y. Zien, J. Y. Zien, Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation, ACM Press, 2003, pp. 178–186.

[12] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, J. Kleinberg, Mining the web's link structure, Computer 32 (1999) 60–67. doi:http://dx.doi.org/10.1109/2.781636.
URL http://dx.doi.org/10.1109/2.781636

[13] M.-M. Naing, E.-P. Lim, R. H. L. Chiang, Extracting link chains of relationship instances from a web site, J. Am. Soc. Inf. Sci. Technol. 57 (2006) 1590–1605. doi:10.1002/asi.v57:12.
URL http://portal.acm.org/citation.cfm?id=1165013