Structural Web Search Using a Graph-Based Discovery System

Nitish Manocha, Diane J. Cook, and Lawrence B. Holder
Department of Computer Science and Engineering
Box 19015
University of Texas at Arlington
Arlington, TX 76019-0015
{cook,holder}@cse.uta.edu

Abstract

Accessing information of interest on the Internet presents a challenge to scientists and analysts, particularly if the desired information is structural in nature. Our goal is to design a structural search engine which uses the hyperlink structure of the Web, in addition to textual information, to search for sites of interest. To design a structural search engine, we use the SUBDUE graph-based discovery tool. The tool, called WebSUBDUE, is enhanced by WordNet features that allow the engine to search for synonym terms. Our search engine retrieves sites corresponding to structures formed by graph-based user queries. We demonstrate the approach on a number of structural web queries.

Introduction

The World Wide Web (WWW) provides an immense source of information. Web search engines sift through keywords collected from millions of pages when responding to a particular query. These search engines typically ignore the importance of searching the structure of hyperlink information and cannot resolve complicated queries like '*find all pages that link to a page containing term x*'. A hyperlink from one page to another represents an organization that a web page designer wants to establish, or relationships that exist between web sites. A structural search engine searches not only for particular words or topics but also for a desired hyperlink structure.

The WWW can be represented as a labeled graph. SUBDUE is a data mining tool that discovers repetitive substructures in graph-based data [Cook et al. 2000]. We hypothesize that SUBDUE can form the heart of a structural search engine, called WebSUBDUE. In particular, this graph-based data mining tool can discover structure formed by a user query within a graph representation of the WWW.

To demonstrate the use of WebSUBDUE as a structural search engine, we perform various experimental queries on the Computer Science and Engineering Department of the UTA web site and compare the results with those of posting similar queries to a popular search engine, Altavista.

In response to a user query, a structural search engine searches not only the text but also the structure formed by the set of pages to find matches to the query. The web consists of hyperlinks that connect one page to another, and this hyperlink structure represents a relationship between the source and destination pages. Such relationships may include a hierarchical relation between a top-level page and a child page containing more detailed information, connections between current, previous and next sites in a sequence of pages, or an implicit endorsement of a site which represents an authority on a particular topic. A structural search engine utilizes this hyperlink structure along with the textual content to find sites of interest. Structural web queries can locate online presentations, web communities, or nepotistic sites. By searching the organization of a set of pages as well as the content, a search engine can provide richer information on the content of web information.

Even a purely textual search can face challenges due to the fact that human language is very expressive and full of synonymy. A query for 'automobile', for example, may miss a deluge of pages that include the term 'car'. One strategy to overcome this limitation is to inform search algorithms about semantic relations between words. A structural search engine could use this information to improve the retrieval of web sites that reflect the structure and the semantics of the user query.

Much research has focused on using hyperlink information in some method to enhance web search. The Clever system scans the most authoritative sites on topics by making use of hyperlinks [Clever]. A respected authority is a page that is referenced by many good hubs; a useful hub is a location that points to many valuable authorities. Google is another search engine that uses the power of hyperlinks to search the web [Brin and Page 1999]. Google evaluates a site by summing the scores of other locations pointing to the page.

Although these systems use hyperlink structures to rank retrieved web pages, they do not perform searches for a range of structural queries. In contrast, WebSUBDUE searches for any type of query describing structure embedded with textual content.

Data Preparation

Viewed as a graph, the World Wide Web structure exhibits properties that are of interest to a number of researchers [Kleinberg et al. 1999]. For this project, we transform web data to a labeled graph for input to the WebSUBDUE system. Data collection is performed using a web robot written in Perl. The web robot only follows links to pages residing on specified servers. As it traverses a web site, the web robot generates a graph file representing the specified site. URLs are processed using a breadth-first search until all pages on the site have been visited or the user-specified page limit is exceeded.

Once the URLs have been scanned and the associative array has been created, a labeled graph is generated representing the website. In this graph, each URL is represented as a vertex labeled '*_page_*', and an edge labeled '*_hyperlink_*' points from parent URLs to child URLs. If the document currently being processed is either an HTML file or a text file, the file is processed to obtain keyword information. Vertices labeled with the corresponding words are added to the graph, and an edge labeled '*_word_*' connects the page in which a keyword is found to the word vertex.

The current search engine allows the user to create a graph for a new domain or search an existing graph. New web sites can also be incrementally added to an existing graph.

SUBDUE

SUBDUE is a knowledge discovery system that discovers patterns in structural data. SUBDUE accepts data in the form of a labeled graph and performs various types of data mining on the graph. As an unsupervised discovery algorithm, SUBDUE discovers repetitive patterns, or subgraphs, in the graph. As a concept learner, the system identifies concepts that describe structural data in one class and exclude data in other classes. The system can also discover structural hierarchical conceptual clusters. Here, we review approach to unsupervised discovery. We then introduce the application of SUBDUE as a structural search engine, called WebSUBDUE, in which the search query and the WWW are represented as labeled graphs and discovered instances are reported as the results of the query.

SUBDUE uses a polynomial-time beam search to find the subgraph which yields the best compression of the input graph. A substructure in SUBDUE consists of a subgraph defining the substructure and all of the substructure instances throughout the graph. The initial state of the search is the set of subgraphs consisting of all uniquely labeled vertices. The only search operator is the ***Extend Subgraph*** operator, which extends a subgraph in all possible ways by a single edge and a neighboring vertex. Substructures are kept on an ordered queue of length determined by the beam width, and are evaluated by their ability to compress the graph, following the Minimum DescriptionLength principle.

Once the search terminates and returns the list of best subgraphs, the graph can be compressed using the best subgraph. The compression procedure replaces all instances of the subgraph in the input graph by a single representative vertex. SUBDUE can iterate on this compressed graph to generate a hierarchical description of discovered substructures.

SUBDUE supports biasing the discovery process toward specified types of substructures. Predefined substructures can be provided as input by creating a predefined substructure graph that is stored in a separate file. SUBDUE will locate and expand these substructures, thus "jump-starting" the discovery process. The inclusion of background knowledge has been shown to be of benefit when expert-supplied knowledge is available [Cook and Holder 1999].

For our structural search engine, WebSUBDUE invokes SUBDUE in predefined substructure mode, where the search query is represented in the form of a substructure. SUBDUE discovers the instances of the predefined substructure in the graph, representing the results of the query. WebSUBDUE reports the graph vertices, edges, and corresponding URLs for each discovered instance.

In many databases, patterns exist in the database with small variations between instances. SUBDUE applies a polynomial-time inexact match algorithm to allow for small differences between a substructure definition and a substructure instance. The dissimilarity of two graphs is determined by the number of transformations needed to make one graph isomorphic to the other. Allowed transformations include adding or delete an edge or vertex, changing a label, and reversing the direction of an edge. Two graphs match if the number of transformations is no more than a user-defined threshold value multiplied by the size of the larger graph. If the threshold value is defined as 0, an exact match is required. SUBDUE's inexact graph match algorithm can be used by WebSUBDUE to find web sites that closely, but not exactly, match the user query.

WordNet

WordNet is an electronic lexicon database which attempts to organize information according to the meanings of the words instead of the forms of the words [Miller et al. 1991]. In this project, our goal is to augment the structural search technique with stored information about relations between words. In particular, WebSUBDUE uses WordNet to increase its search capacity by searching for words similar to the query term rather than searching only for the query term itself.

We enhance the capabilities of WebSUBDUE's structural search engine by integrating some of WordNet's functionality. WebSUBDUE uses WordNet to enhance its search capabilities by finding words similar to the search terms. In particular, vertices representing a word within a page are matched with vertices containing a different label if they hold one of the valid relationships defined by WordNet. Valid considerations include matching the base form of the word (e.g., match '*base*' or '*basis*' with query

'*bases*'), derived forms of the word (e.g., match '*Jan*' with query '*January*'), and synonyms (e.g., match '*employment*', '*position*', or '*work*' with query '*job*').

Experimental Results

We conducted a number of experiments to evaluate the capabilities of the WebSUBDUE structural search engine. Where possible, we compare query results of WebSUBDUE with search results generated using a popular search engine, Altavista. Altavista's advanced search features include the use of Boolean expressions, the limitation of a search to a particular host or domain, and other search criteria that provide a valuable point of comparison for the results discovered by WebSUBDUE.

In these experiments, we restrict the search space to a single sample domain. The sample domain chosen is the web domain for the UTA CSE department. A graph file of the CSE-UTA domain, http://www-cse.uta.edu, was created using the web robot. The graph representation of the CSE-UTA web site contains 113,933 vertices and 125,657 edges covering 5,825 URLs.

To demonstrate the structural searching capability of WebSUBDUE, we posed several queries to the system and compared the results to similar searches using Altavista. The search range of Altavista was also restricted to the CSE-UTA domain using the define-host advanced search option in Altavista, thus we eliminate the domain name from our summaries.

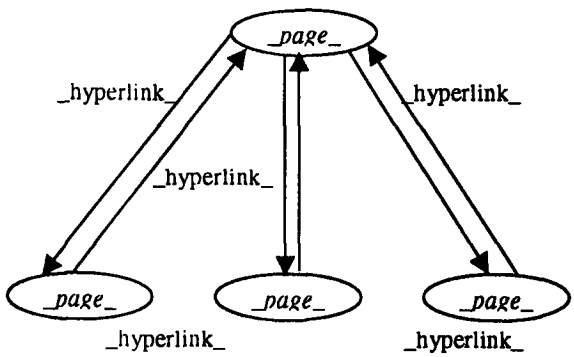
Searching for Presentation Pages

One type of structural query might be to find online lecture materials or HTML papers discussing a particular topic. A query to search for all presentation-style pages was posed to WebSUBDUE. A presentation page is defined here as series of URLs with links to the *next* page, the *previous* page and the *top* page, the top page being a menu page which has links to all the other pages. This query structure is shown in Figure 1.

In response to the presentation page query, WebSUBDUE discovered 22 unique instances or 22 presentation pages on the CSE-UTA site. The discovered URLs are listed in Figure 1. These discovered web pages represent presentation pages covering various topics with a top menu page containing a link to sub-topic pages and each sub topic page pointing to the next sub topic page, the previous page and the top menu page.

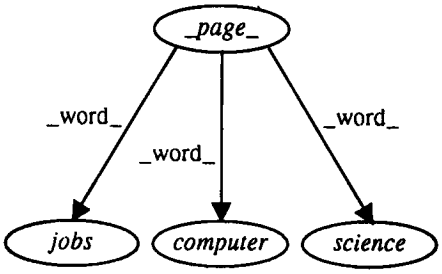
The Altavista search engine has no method of responding to structural search queries. To assist Altavista, we provided additional information to guide Altavista. Altavista was run in the advanced search mode with the following option: *"host:www-cse.uta.edu AND image:next_motif.gif AND image:up_motif.gif AND image:previous_motif.gif."* This search option asks Altavista to search in the CSE-UTA domain for pages containing links to next icon, previous icon, and top icon files.

Altavista discovered 12 instances compared to the 22 instances WebSUBDUE discovered. Because the actual names of the icons varied between presentations, not all presentation pages were discovered using Altavista. This shows the difficulty of posing these types of structural queries using a text search engine, whereas WebSUBDUE just requires the structure information.



1. ~holder/courses/cse5311/lectures/l26/l26.html
2. ~cook/ai1/syllabus/syllabus.html
3. ~holder/courses/cse2320/lectures/l3/l3.html
4. ~holder/courses/cse5311/lectures/l12/l12.html
5. ~holder/courses/cse2320/lectures/l16/l16.html
6. ~holder/courses/cse2320/lectures/l10/l10.html
7. ~holder/courses/cse5311/lectures/l19/l19.html
8. ~holder/courses/cse5311/lectures/l9/l9.html
9. ~cook/ai1/lectures/l13/l13.html
10. ~cook/ai1/lectures/l1/l1.html
11. ~reyes/node48.html
12. ~reyes/node1.html
13. ~cook/ai1/lectures/l6/node6.html
14. ~cook/ai1/lectures/l7/l7.html
15. ~holder/courses/cse5311/lectures/l23/l23.html
16. ~holder/courses/cse2320/lectures/l2/l2.html
17. ~cook/ai1/lectures/l10/l10.html
18. ~cook/ai1/lectures/l12/l12.html
19. ~holder/courses/cse5311/lectures/l10/l10.html
20. ~holder/courses/cse2320/lectures/l15/l15.html
21. ~holder/courses/cse5311/lectures/l4/l4.html
22. ~holder/courses/cse5311/lectures/l2/l2.html

Figure 1. Presentation page structural query.



WebSUBDUE results

1. about_uta_engr.html (employment)
2. Undergrad/undergrad.html (work)
3. ~cook/ai1/hw/h16 (job)
4. ~cook/grt/index.ht ml (job)
5. Undergrad/bscse.html (work)
6. Graduate/mse.html (work)
7. ~cook/pai/pai.html (problem)
8. jobs.html (jobs)
9. ~reyes/node11.html (jobs)
10. ~al-khaiy/cse3320_summer99/summer99.htm (task)

...

30. ~holder/courses/cse5311/summ98/syllabus/syllabus.html (problem)
31. ~cook/aa/syllabus (problem)
32. ~cook/ai1/lectures/applets/pd/ga-axelr.htm (problem)
33. Undergrad/97/97program_info_4.html (problem)

Altavista results

1. Graduate/mse.html
2. jobs.html

Figure 2. Structural query to find pages on 'jobs in computer science'.

Text Searching Using Synonyms

To demonstrate the synonym searching capability of WebSUBDUE, we posed queries requesting all pages on 'jobs in computer science' to WebSUBDUE using the structural query shown in Figure 2 and the synonym match capability, as well as to Altavista. WebSUBDUE discovered 33 instances of the search query, as summarized in Figure 2 along with the match words found in the corresponding page. WebSUBDUE finds *'employment'*, *'work'*, *'job'*, *'problem'*, *'task'* as synonyms and allowable forms of the search terms. Only exact matches were found for *'computer'* and *'science'* in these pages.

A similar query was posed to Altavista. Altavista discovered 2 instances of the search query, as shown in Figure 2. Altavista discovered only 2 sites, both of which were also discovered by WebSUBDUE. WebSUBDUE thus covers a wider search range using synonyms of the query terms along with the keywords themselves.

Inexact match for hubs and authorities

SUBDUE's inexact graph match capability can be used to enhance the WebSUBDUE's web search results. Using an inexact match, web sites can be retrieved which approximate the query structure. The amount of allowable difference is controlled by the user-defined match threshold.

To demonstrate the impact of using an inexact graph match, we search for hub (pages that point to at least three authorities) and authority (pages pointed to by at least three hubs) sites that focus on 'algorithms' with an inexact threshold of 0.2. A total of 13 instances are identified with this query, as opposed to the single instance found using an exact match. Two instances of the query structure shown in Figure 3 are retrieved. The first instance is missing two edges that are provided in the query structure. The second instance is missing three edges and includes an extra node with an associated edge.

Results of these experiments indicate that WebSUBDUE can successfully perform structural search, text search, synonym search, and combinations of these searches. These experiments also demonstrate the need for structural search engines and the inability of existing search engines to perform these functions.

Conclusions

In this paper, we introduce the concept of a structural search engine and demonstrate that web search can be enhanced by searching not only for text but also for structure that has been created to reflect relationships between individual pages. We have demonstrated how WebSUBDUE can be used successfully as a structural search engine, and how the search can be enhanced using WordNet functions and an inexact graph match algorithm.

To improve the textual search capabilities of WebSUBDUE, additional features of WordNet, such as hyponyms and hypernyms, can be integrated. Distances between words can be calculated using techniques such as marker passing [Lytinent et al. 2000] or Latent Semantic Analysis [Landauer et al. 1998] and calculated as part of the graph transformation cost. A user interface that facilitates generation of structural queries is also being considered.

In addition to using WebSUBDUE as a structural search engine, we are also exploring other methods by which SUBDUE can perform data mining on web data. The unsupervised discovery, concept learning, and hierarchical structural clustering capabilities may provide useful insight into the structure and content of web data.

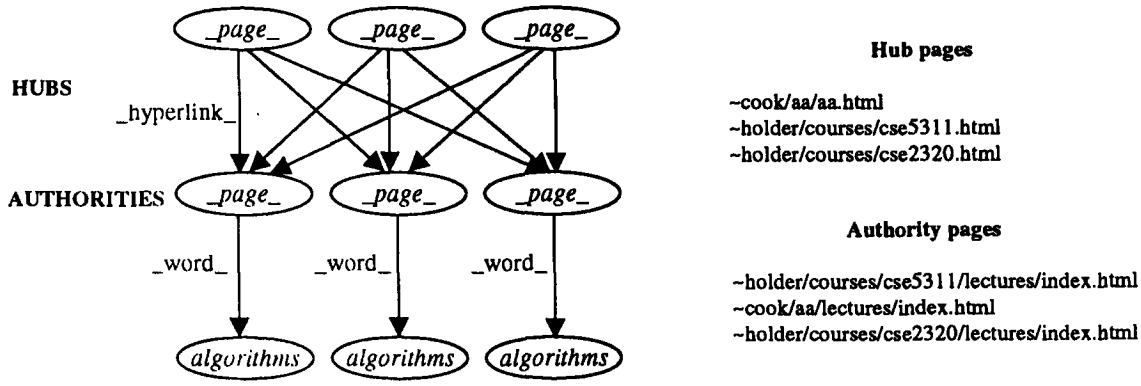


Figure 3. Structural query for hub and authority pages on 'algorithms'.

Acknowledgements

This work was supported by NSF grant EIA-0086260.

References

Brin S., and Page, L. 1999. The Anatomy of a Large-scale Hypertextual Web-search Engine. In *Proceedings of the Seventh International World Wide Web Conference.*

http://www.almaden.ibm.com/cs/k53/clever.html.

Chakrabarti, S., Dom, B. E., Gibson, D., Kleinberg, J., Kumar, R., Raghavan, P., Rajapolan, S., and Tompkins, A. 1999. Mining the Link Structure of the World Wide Web. *IEEE Computer*, 32(8), 60-67.

Cook, D. J., Holder, L. B., Galal, G., and Maglothin, R. 2000. Approaches to Parallel Graph-Based Knowledge Discovery. *Journal of Parallel and Distributed Computing.*

Cook, D. J. and Holder, L. B. 2000. Graph-Based Data Mining, *IEEE Intelligent Systems*, 15(2), 32-41.

Landauer, T. K., Foltz, P. W., and Laham, D. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.

Lytinen, S. L., Tomuro, N., and Repede, T. The Use of WordNet Sense Tagging in FAQFinder. In *Proceedings of the AAAI Workshop on Artificial Intelligence for Web Search.*

Miller, G. A. Beckwith, R., Fellbaum, C., Gross D., and Miller, K. J. 1991. Introduction to WordNet: An On-line Lexical Database, *International Journal of Lexicography*, 3(4), 235-244.