# Knowledge Representation, Learning, and Reasoning in WebDoc - A Web Document Classification System

**Bo Tang and Julia Hodges**

Department of Computer Science
Mississippi State University
Box 9637
Mississippi State, MS 39762-9637
btang@cs.msstate.edu, hodges@cs.msstate.edu

### Abstract

This paper describe a novel approach to knowledge representation, learning, and reasoning in WebDoc, a system that classifies Web documents according to the Library of Congress classification system. We argue that an automatically constructed domain-independent knowledge base is indispensable. The WebDoc system builds a knowledge base (represented as a semantic network) that contains the Library of Congress subject headings and their relationships. Through training on human-indexed and NLP-parsed Web documents, WebDoc modifies the semantic network and generates rules for future index generation tasks.

## Introduction

The rapid growth of the World Wide Web makes a tremendous amount of information available to people who have access to a computer connected to the Internet. However, there is still a long way to go from simply having access to really taking advantage of the information. People often get lost rather than enlightened due to the lack of efficient Web information retrieval tools.

One of the important tasks of information retrieval is indexing because it produces a set of representatives, i.e., indexes, for the contents of each document, thus facilitating the process of classification. In automatic indexing, there are many differences between the traditional information retrieval systems and today's Web-based information retrieval system. In fact, Web document retrieval and classification is believed to be much more difficult because of the dynamic and anarchical nature of the Web.

Web documents often feature free-style writing. Moreover, geographically distributed amateur writers differ greatly in how they express the same thing. Some of them might even use different languages. Therefore, the vocabulary mismatch problem caused by the discrepancy between the phrases in the texts and index phrases is even worse when dealing with Web documents. A common approach to resolving the vocabulary mismatch problem is to build a thesaurus and map terms in the texts to their conceptual counterparts. The higher-level abstraction will help bridge the discrepancies in the lower lexical level. Because of Web documents' diversity and size, it is impossible to build a thesaurus manually. Therefore, an automatically constructed domain-independent knowledge base is indispensable.

In the rest of this paper, we will first discuss some related work. Then an overview of the WebDoc system will be given, followed by a discussion of the methodologies applied to knowledge representation, rule generation, and reasoning in the system. Finally, some conclusions will be drawn and possible future work will be proposed.

## Related Work

This research has evolved from an earlier project, AIMS (Assisted Indexing at Mississippi State), which involved the development of an automated system to aid human document analysts in the assignment of indexes to physical chemistry journal articles (Hodges et al. 1996; Hodges et al. 1997). In the AIMS project, the problem was confined to a small domain and the documents processed by the system were restricted to journal articles, which are usually in much more uniform format than Web documents. A small domain allows human experts to build the knowledge base for the system manually.

Some Web search engines, such as Yahoo, maintain a hierarchical classification system and return retrieved documents along with their associated categories. Others simply return a flat ordered list of documents.

One of the problems most of the search engines have is the lack of precision in their output. Usually a search engine will return hundreds of documents for a particular user query. Though most of the search results are presented as ordered lists in which documents are ranked based on their similarity to the query, it is very hard for users to sift through that many documents and find the ones they want. In order to tackle this problem, some post-query techniques are created to organize returned documents in a more informative way. Some systems use document clustering to create groups of documents based on the associations among the documents (Hagen 1997; Allen, Obry, and Littman 1993; Hearst and Pedersen 1996). A popular search engine, Northern Light, uses a hierarchical clustering to help users locate the desired Web pages. There are also some systems using domain knowledge to build a hierarchical system and then classify returned documents into different categories (Pratt, Hearst, and Fagan 1999).

A common component of the knowledge base in most information retrieval systems is a thesaurus. Much research has been done on how to construct a thesaurus automatically (Spark Jones and Needham 1968; Salton 1968; Rijsbergen, Harper, and Porter 1981). However, the usage of thesauri was single-purposed, i.e., to decrease a term's specificity and therefore improve the recall rate of the retrieval system. The quantitative aspect of the relationships between terms was not taken into consideration.

Jing and Croft (1994) proposed a thesaurus construction approach in which noun phrases instead of words are used as the main characteristic features because evidence shows that noun phrases characterize the content of a text better than words (Croft, Turtle, and Lewis 1991). In our research, we adopt a similar approach, i.e., we use noun phrases as the main characteristic features of the text.

An inference network based probabilistic retrieval model has been proposed to combine multiple sources of evidence about document and query in order to decide if a given document matches an information need (Turtle and Croft 1990). In our research, we use a certainty factor approach to combine multiple sources of evidence.

## System Overview

The major task of WebDoc is to build a knowledge base and then classify Web documents by consulting the knowledge base. There are three major components in the WebDoc system: the NLP component, the knowledge base construction component, and the index generation component. They correspond to three important phases of WebDoc's working process: data preparation, training, and indexing.

Web documents are downloaded to serve as an input to the WebDoc system. First, the NLP (natural language processing) component tags the text with syntactic information and parses the documents in order to identify noun phrases. Once noun phrases have been identified, they are sent to the knowledge base construction component, where statistical and thesaurus information is extracted and stored in the knowledge base. After a large number of Web documents have been used to construct the knowledge base, the system finishes training and starts to perform the task of indexing. The index generation component greatly relies on the knowledge base to perform its function. The overall architecture of the WebDoc system is illustrated in Figure 1.

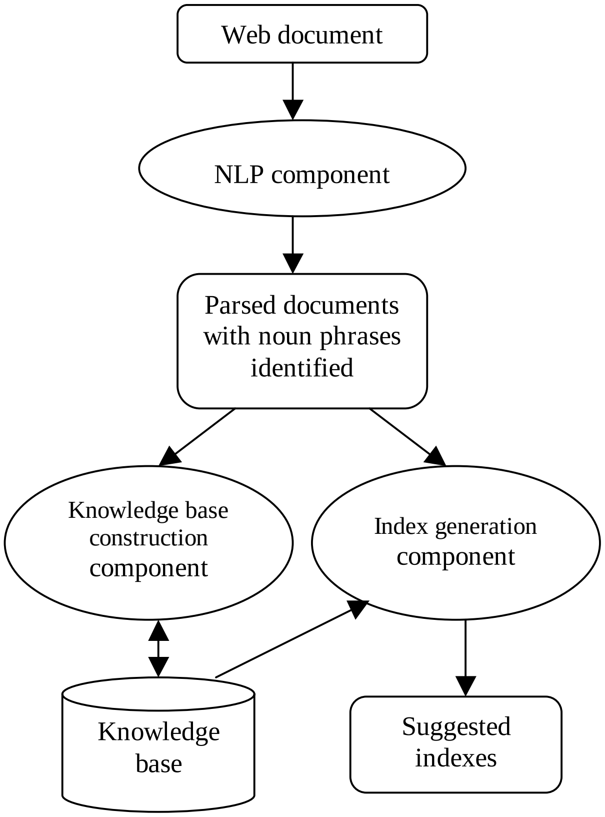

Web document
NLP component
Parsed documents with noun phrases identified
Knowledge base construction component
Index generation component
Knowledge base
Suggested indexes

Figure 1. The Architecture of the WebDoc System

## Methodology

The major task of the WebDoc system is to extract knowledge from training documents and represent it in a way that the index generation process can benefit from. It is also very important for us to decide how to utilize the knowledge stored in the knowledge base. Below, we discuss the representation, learning, and reasoning of knowledge in the WebDoc system.

### Knowledge Representation

The foundation of the knowledge base in the WebDoc system is the Library of Congress subject headings (LCSHs). There are many relationships between LCSHs that we can put into the knowledge base. One LCSH might have a group of LCSHs as its narrower topics (NT) or broader topics (BT), roughly constituting a hierarchical relationship. A LCSH may have a more preferred term (labeled USE), a set of less preferred terms (labeled UF), or some related topics (RT) that constitute a loose equivalence relationship. Note that USE and UF are pairs of inverse relationships.

Based on all these relationships, we build a thesaurus where related concepts are put into a common thesaurus class. A thesaurus class is organized as a semantic network in which each node represents a concept and links represent relationships between concepts. A part of the thesaurus class that contains *Web search engines* is shown in Figure 2.

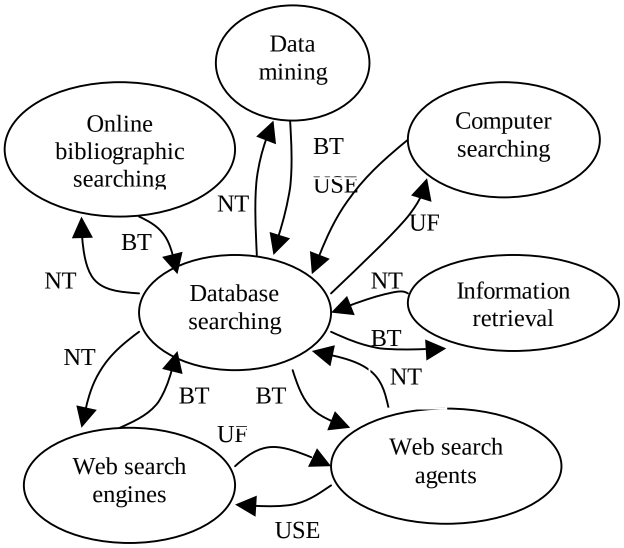


Figure 2. Example of a Thesaurus Class (Partial)

Once an initial version of the thesaurus is acquired from the LCSHs, the knowledge base construction component begins to extract relevance patterns between concepts in the Web documents and those indexes assigned by human experts. The newly acquired knowledge will be added into the knowledge base by adding concepts and links into the semantic network as the training process continues. Note that the type of relationship between concepts does not play any role in the subsequent index generation process. Therefore there is no need to assign a label for newly added links.

## Rule Extraction

Two statistics, *good hits* and *bad hits*, are collected to extract rules for indexing. *Good hits* is the number of times that a particular concept is indexed to a specific index. *Bad hits* is the number of times that a particular concept is not indexed to a specific index. *Good hits* and *bad hits* are used to calculate the conditional probability of a particular concept being indexed to a specific index.

Suppose *a* and *b* are two concepts in the knowledge base and there is a link in the semantic network from *a* to *b* representing some sort of relationship. Through training, we get *good_hits (a, b)* and *bad_hits(a, b)* for this link, representing the number of times that *b* is indexed and the number of times that *b* is not indexed, respectively, given that *a* appears in a document. A certainty factor (CF) rule can be extracted here. If *B* represents the hypothesis that *b* is indexed and *A* represents the evidence that *a* is observed in the text, the degree of belief in hypothesis *B* when evidence *A* is observed can be calculated as:

$$CF(B, A) = \frac{good\_hits(a,b)}{good\_hits(a,b) + bad\_hits(a,b)}$$

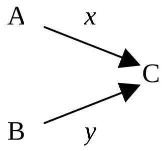
In other words, the CF of hypothesis *B* confirmed by evidence *A* is approximately equal to its conditional probability for that evidence (Stefik 1995).

There is always a possibility that a concept can be inferenced by itself. Therefore a corresponding rule is extracted for each self-inferenced concept.

## Reasoning and Index Generation

Once the knowledge base is trained, the WebDoc system assigns indexes to the Web documents. In the data preparation process, a Web document has its noun phrases recognized. As soon as the concepts in the document are identified, the system must decide which Library of Congress subject heading is a good indicator of the document contents and hence should be indexed. In order to do this, the WebDoc system assigns weights to those concepts that can be inferenced from concepts in the Web documents. The weight assigned to each candidate concepts is essentially its certainty factor.

Most likely, a concept will be inferenced by multiple rules, or by a single rule several times. Multiple sources of evidence need to be combined to calculate the final belief for a hypothesis. Though these evidences are not conditionally independent, it would be difficult to accurately estimate the prior and conditional probabilities required to use Bayes' rule due to the lack of sufficient data. Therefore we assume the conditional independency of multiple evidence and use the calculus of certainty combination to calculate the certainty factor for each candidate concept. Since the certainty factors in this application are always greater than 0, the equation for combining two pieces of evidence is:



$$CF(C) = x + y - xy$$

This equation can be used incrementally when new evidence is acquired. That saves a lot of computation cost comparing to the combination equation for an inference network, which requires recalculation of all the evidences once new evidence is acquired.

# Conclusions and Future Work

Currently, we have trained the system on 484 Web documents and tested with another 59 documents. The preliminary tests have produced precision rates that range from 9% to 39% and recall rates that range from 10% to

70%, depending on the cutoff threshold. The results are promising because we are working on a relatively small training set and are training without the semantic tags on noun phrases and other contextual information. We expect the results to improve significantly once we enlarge the training set and add the semantic tags and other contextual information.

## Acknowledgments

The authors would like to thank the National Science Foundation (award # IIS-9734807) for their support of this work. This project is also made possible thanks to donations of data and time from the Online Computer Library Center, Inc. (OCLC), a not-for-profit service and research organization affiliating more than 60,000 libraries worldwide. We would also like to thank Lois Boggess, Lynellen Perry, and Janna Shaffer, who work on the natural language processing part of this project.

## References

Allen, R. B., Obry, P., and Littman, M. 1993. An interface for navigating clustered document sets returned by queries. *In Proceedings of the ACM SIGOIS: Conference on Organizational Computing Systems (COOCS) held at Milpitas, CA*, 166 - 171.

Croft, W. B., Turtle, H. R., and Lewis, D. D. 1991. The use of phrases and structured queries in information retrieval. *SIGIR '91*, 32-45

Hagen, E. 1997. *An information retrieval system for performing hierarchical document clustering*. Dartmouth College: Department of Computer Science. Technical Report PCS-TR97-318.

Hearst, M. A. and Pedersen, J. O. 1996. Reexamining the cluster hypothesis: Scatter / Gather on retrieval results. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR '96),* 76 - 84.

Hodges, J., Yie, S., Reighart, R., and Boggess, L. 1996. An automated system that assists in the generation of document indexes. *Natural Language Engineering* 2 (2): 137 - 160.

Hodges, J., Yie, S., Kulkarni, S., and Reighart, R. 1997. Generation and evaluation of indexes for chemistry articles. *Journal of Intelligent Information Systems* 7: 57 - 76.

Jing, Y. and Croft, W.B. 1994. *An association thesaurus for information retrieval.* University of Massachusetts at Amherst: Department of Computer Science. Technical Report 94-17.

Pratt, W., Hearst, M. A., and Fagan, L. M. 1999. A knowledge-based approach to organizing retrieval documents. In *Proceedings of the 16th National Conference on AI (AAAI '99) held at Orlando, Florida, July 1999.*

Rijsbergen, C. J., Harper, D. J., and Porter, M. F. 1981. The selection of good search terms. *Information Processing and Management* 17: 77 – 91.

Salton, G. 1968. *Automatic Information Organization and Retrieval*. McGraw-Hill, Inc.

Spark Jones, K., and Needham, R. M. 1968. Automatic term classification and retrieval. *Information Processing and Management* 4: 91 – 100.

Stefik, M. 1995. *Introduction to Knowledge System*. Morgan Kaufmann Publishers, Inc.

Turtle, H. R., and Croft, W. B. 1990. Inference networks for document retrieval. *SIGIR '90*, 1-24.