# Spidering and Filtering Web Pages for Vertical Search Engines

**Michael Chau**
**The University of Arizona**
mchau@bpa.arizona.edu

## 1 Introduction

The size of the Web is growing exponentially. The number of indexable pages on the web has exceeded 2 billion (Lyman & Varian, 2000). It is more difficult for search engines to keep an up-to-date and comprehensive search index, resulting in low precision and low recall rates. Users often find it difficult to search for useful and high-quality information on the Web. Domain-specific search engines (or vertical search engines) alleviate the problem to some extent, by allowing users to perform searches in a particular domain and providing customized features. However, it is not easy to build these search engines. There are two major challenges to building vertical search engines: (1) Locating relevant documents from the Web; (2) Filtering irrelevant documents from a collection. This study tries to address these issues and propose new approaches to the problems.

## 2 Research Background

### 2.1 Creating Vertical Search Engines

Web search engines such as Google and AltaVista provide the most popular way to look for information on the Web. Many users begin their Web activities by submitting a query to a search engine. However, because of the enormous size of the Web, general-purpose search engines can no longer satisfy the needs of most users searching for specific information on a given topic. Many vertical search engines, or domain-specific search engines, have been built to facilitate more efficient search in various domains. These search engines usually provide more precise results and more customizable functions. For example, *LawCrawler* (www.lawcrawler.com) allows users to search for legal information. *BuildingOnline* (www.buildingonline.com), *BioView.com* (www.bioview.com), and *SciSeek* (www.sciseek.com) are a few other examples.

As mentioned previously, there are two main problems in building vertical search engines:

- The search engine spider needs to predict whether a URL points to a relevant Web page to improve efficiency.
- The search engine needs to determine the content and quality of Web pages to avoid irrelevant or low-quality ones.

Search engines usually use spiders (also referred to as Web robots, crawlers, worms, or wanderers) to retrieve pages from the Web by recursively following URL links in pages using standard HTTP protocols. These spiders use different algorithms to control their search. The following methods are commonly used to locate Web pages relevant to a particular domain:

- The spiders can be restricted to stay in particular Web domains, because many Web domains have specialized contents (Manber et al., 1997; Witten et al., 2001). For example, most Web pages within the domain *www.toyota.com* will be relevant to automobiles.
- Some spiders are restricted to collect only pages at most a fixed number of links away from the starting URLs or starting domains (e.g., Manber et al., 1997; Sumner et al., 1998).
- More sophisticated spiders analyze Web pages and hyperlinks to decide what documents should be downloaded (Cho et al., 1998; McCallum et al., 1999).

These methods have different levels of performance in efficiency and effectiveness, but in most cases the resulting collection is still noisy and needs further processing. Filtering programs are needed to filter irrelevant and low-quality pages from the collection to be used in the vertical search engine. Surprisingly, it appears that many vertical search engines do not perform

filtering (e.g., NanoSpot, Galaxy). They assume that most pages found in the starting domains (or at a specified depth) are relevant. For search engines that employ filtering, the techniques used include:

- Domain experts manually determine the relevance of each Web page (e.g., Yahoo).
- In the simplest automatic way, the relevance of a Web page can be determined by the occurrences of particular keywords (e.g., computer) (Cho et al., 1998).
- TF*IDF (term frequency * inverse document frequency) is calculated based on domain-expert created lexicon. Web pages are compared with a set of relevant documents, and those with a similarity score above a certain threshold are considered relevant (Baujard et al., 1998).
- Text classification techniques, such as Naive Bayesian classifier, also have been applied to Web page filtering (Chakrabarti et al., 1998, McCallum et al., 1999).

## 2.2 Text Classification

Text classification is the study of classifying textual documents into predefined categories. The topic has been extensively studied at SIGIR conferences and evaluated on standard testbeds. There are a few major approaches. For example, the Naive Bayesian method has been widely used (e.g., Koller & Sahami, 1997; Lewis & Ringuette, 1994; McCallum et al., 1999). It uses the joint probabilities of words and categories to estimate the probabilities of categories given a document. Documents with a probability above a certain threshold are considered relevant.
The k-nearest neighbor method is another widely used approach in text classification. For a given document, the k neighbors that are most similar to a given document are first identified (Iwayama & Tokunaga, 1995; Masand et al., 1992). The categories of these neighbors are then used to decide the category of the given document. A threshold is also used for each category.

Neural network programs, designed to model the human neural system and learns patterns by modifying the weights among nodes based on learning examples, also have been applied to text classification. Feedforward/backpropagation neural network was usually used (Wiener et al., 1995; Ng et al., 1997; Lam & Lee, 1999). Term frequencies or TF*IDF of the terms are used as the input to the network. Based on learning examples, the network will be trained to predict the category of a document.
Another new technique used in text classification is called support vector machine (SVM), a statistical method that tries to find a hyperplane that best separates two classes (Vapnik, 1995). Joachims first applied SVM in text classification problem (Joachims, 1998). It has been shown that SVM achieved the best performance on the Reuters-21578 data set (Yang and Liu, 1999).

In addition to general text documents, classification also has been studied on Web pages. Web pages are often noisy, but they provide additional information for each document. For example, text from neighborhood documents has been used in attempt to improve classification performance. However, it turns out to worsen performance because there are often too many neighbor terms and too many cross-linkages between different classes (Chakrabarti et al., 1998; Yang et al., 2002). Use of other information of neighborhood documents have been proposed, such as the predicted category of neighbors as additional information (Chakrabarti et al., 1998; Oh et al., 2000), the anchor text pointing to a document (Furnkranz, 1999), or the outgoing links to all other documents (Joachims et al., 2001). It has been shown that using such additional information improves classification results.

## 2.3 Web Structure Analysis

In recent years, Web link structure has been widely used to infer important information about pages, intuitively, the author of a Web page A places a link to Web page B if he or she believes that B is relevant to A, or of good quality. Usually, the larger the number of in-links, the better a page is considered to be. The rationale is that a page referenced by more people is likely to be more important than a page that is seldom referenced. One can also obtain the anchor text that describes a link. Anchor text is the underlined, clickable text of an outgoing link in a Web page. Anchor text may provide a good description of the target page because it represents how other people linking to the page actually describe it.
In addition, it is reasonable to give a link from an authoritative source (such as Yahoo) a higher weight than a link from an unimportant personal homepage. Researchers have developed several algorithms to address this issue. Among these, PageRank and HITS are the two most widely used.

The PageRank algorithm is computed by weighting each in-link to a page proportionally to the quality of the page containing the in-link (Brin & Page, 1998). The quality of these referring pages also are determined by PageRank. Thus, the PageRank of a page p is calculated recursively as follows:

$$PageRank(p) = \frac{1-d}{n} + d \times \sum_{\substack{all\ q\ linking \\ to\ p}} \left( \frac{PageRank(q)}{c(q)} \right)$$

where *d is a damping factor between 0 and 1,*
*n is the total number of pages in the collection,*
*c*(*q*) *is the number of out-going links in q.*

Intuitively, a Web page has a high PageRank score if the page is linked from many other pages, and the scores will be even higher if these referring pages are also good pages (pages that have high PageRank scores). It is also interesting to note that the PageRank algorithm follows a random walk model. The PageRank score of a page is proportional to the probability that a random surfer clicking on random links will arrive at that page. Applied in the commercial search engine Google, this score has been shown to be very effective for ranking search results (Brin & Page, 1998). Computation time, however, is a main problem in using PageRank. The PageRank score of each Web Page has to be calculated iteratively, making it computationally expensive.

Kleinberg proposed a measure called the HITS (Hyperlink-Induced Topic Search) algorithm (Kleinberg, 1998), which is similar to PageRank, in the HITS algorithm, authority pages are defined as high-quality pages related to a particular topic or search query. Hub pages are those that are not necessarily authority themselves but provide pointers to other authority pages. A page that many others point to should be a good authority, and a page that points to many others should be a good hub. Based on this intuition, two scores are calculated in the HITS algorithm for each Web page: an authority score and a hub score. They are calculated as follows:

$$AuthorityScore(p) = \sum_{\substack{all\ q\ linking \\ to\ p}} \left( HubScore(q) \right)$$

$$HubScore(p) = \sum_{\substack{all\ r\ linking \\ from\ p}} \left( AuthorityScore(r) \right)$$

A page with a high authority score is one pointed to by many hubs, and a page with a high hub score is one that points to many authorities. One example that applies the HITS algorithm is the Clever search engine (Chakrabarti et al., 1999) which achieves a higher user evaluation than the manually compiled directory of Yahoo.

## 3 Research Questions

Based on the review, I found it possible to improve the traditional approaches used in building vertical search engines described in Section 2.1. In Web page spidering, existing approaches are either computationally expensive or not effective. The performance would be improved if Web content and Web structure analysis could be effectively combined. In Web page filtering, most traditional approaches are not robust to spamming, a popular practice in which Web page authors manipulate the page content to boost ranking. Using text classifiers for Web page filtering seem to be the most promising, given their good performance in traditional text classification. However, a major problem is that most classifiers were evaluated using at least 2/3 data for training, which is not feasible in vertical search engine creation as only a small number of documents (hundreds) can be tagged for classifying a large number of documents (millions). Also, in both Web page spidering and Web page filtering, most existing techniques do not make use of domain knowledge, and they do not consider the quality of documents, which is important in vertical search engines.

In this study, the following research questions are investigated: (1) Can existing graph search algorithms and Web analysis techniques be integrated to create a collection of Web pages of high relevance and good quality for a vertical search engine? (2) Can Web structure analysis techniques be used to help create a vertical search engine? (3) Can domain knowledge be used to enhance Web page spideirng and Web page filtering for a vertical search engine? (4) Can Web page classification be applied to a large collection (e.g., a million documents) with only a standard number (a few hundred) of training examples?

# 4 Proposed Approach

## 4.1 Web Page Spidering

Aimed at combining different Web content and structure analysis techniques to build spider programs for vertical search engines, we developed and compared three versions of Web spiders, namely, (1) Breadth-First Search (BFS) Spider, (2) PageRank Spider, and (3) Hopfield Net Spider. The BFS Spider follows a simple breadth-first search algorithm. The PageRank Spider, adopted from Cho et al. (1998), employs a best-first search, using the PageRank score of each unvisited URL as the heuristics. The Hopfield Net Spider follows a spreading activation algorithm. In this approach the Web is modeled as a Hopfield Net, a single-layered, weighted neural network (Hopfield, 1982). Nodes are activated in parallel and activation values from different sources are combined for each individual node until the activation scores of nodes on the network reach a stable state (convergence). The Hopfield Net Spider starts with a set of seed URLs represented as nodes, activates neighboring URLs, combines weighted links, and determines the weights of newly discovered nodes. An initial set of seed URLs is given to the system and each of them is represented as a node with a weight of 1. $\mu_i(t)$ is defined as the weight of node $i$ at iteration $t$. The spider fetches and analyzes these seed Web pages in iteration 0. The new URLs found in these pages are added to the network. Proceeding to the next iteration, the weight of each node is calculated as follows:

$$\mu_i(t+1) = f_s\left(\sum_{\substack{every\ known\\ parent\ \ h\ of\ i}} w_{h,i}\,\mu_h(t)\right)$$

where $w_{h,i}$ is the weight of the link between two nodes and $f_s$ is the SIGMOID transformation function that normalized the weight to a value between 0 and 1. After the weights of all the nodes in the current iteration are calculated, the spider needs to decide which node (URL) should be activated (visited) first. As the weights decide the order in which URLs are to be visited, it is very critical to the effectiveness of the algorithm. The set of nodes in the current iteration are then visited and fetched from the Web in descending order of weight. After all the pages with a weight greater than a threshold have been visited and downloaded, the weight of each node in the new iteration is updated to reflect the quality and relevance of the downloaded page content. The above process is repeated until the required number of Web pages have been collected or until the average weight of all nodes in an iteration is smaller than a maximum allowable error.

## 4.2 Web Page Filtering

In order to address the problems with the current approaches in Web page filtering, feature-based approach is proposed, instead of representing each document as a bag of words, each Web page is represented by a limited number of features. This reduces the dimensionality of the classifier and thus the number of training examples needed. The characteristics of Web structure also can be incorporated into these features.

The relevance and quality of a Web page can be determined by the following aspects: (1) page content, (2) page content of neighbors, and (3) link information. Several features are defined for each aspect. Page content score can be defined based on a domain lexicon (Baujard et al., 1998). Two features will be used:

1. Title($p$) = Number of terms in the title of page p found in domain lexicon
2. TFIDF($p$) = Sum of TFIDF of the terms in page p found in domain lexicon

To incorporate the page content of the neighborsof a page, a score from neighborhood documents can be used instead of using all the terms from neighborhood which appear to be more harmful than helpful (Chakrabarti et al., 1998; Yang et al., 2002). Three types of neighbors will be considered: incoming, outgoing, sibling (Chakrabarti et al., 1998). Two content scores (title and TFIDF scores) of the neighborhood documents can be determined similarly to the previous ones. Six features will be used: the averages of the two scores for all incoming neighbors, the averages for all outgoing neighbors, and the averages for all siblings.

Connectivity (link analysis) is used to determine the quality of a page. Link analysis (such as number of in-links, HITS and PageRank) have been useful in many other Web applications such as search result ranking (Brin & Page, 1998; Chakrabarti et al., 1999), but have not been used in text classification. Six scores, namely hub score, authority score, PageRank score, number of in-links, number of out-links, and number of relevant terms in the anchor texts, will be used as features.

The 14 features identified will be used as input to a classifier. A neural network and a SVM are proposed. A feedforward/backpropagation neural network will be adopted because of its robustness and wide usage in classification. The input layer of the neural network will have 14 nodes that correspond to the 14 feature scores of each page. There will be one single output node determining the combination of relevance and quality of a page (whether a Web page should be included in the vertical search engine or not), in order to allow better comparison, a support vector machine also will be used because of its outstanding performance in traditional text classification (Yang & Liu, 1999). It will be used to perform classification based on the same feature scores.

# 5 Evaluation

In order to evaluate the proposed approach, experiments that compare the proposed approaches with traditional approaches have been conducted.

The medical field was chosen as the domain for the evaluation, in the medical domain, it is important for a vast diversity of users (including medical doctors, researchers, librarians and general public) to find important and high-quality information on health topics on the Web. It is also important to distinguish between web pages of good and poor quality. A medical lexicon was created based on the UMLS Metathesaurus, a widely-used medical thesaurus.

A testbed was created by running a random-first search that started with 5 URLs in the medical domain and traversed the Web following random outgoing links. The random-first search was run until 1 million pages were collected and indexed. The testbed represented a typical collection from simple Web spiders, and consisted of 1,040,388 valid, unique Web pages.

We designed and conducted two experiments to compare the three spidering algorithm. Each spider was executed to perform virtual spidering on the testbed. Although the local repository contained information of all the pages, each spider could access only information based on pages it already had visited. The same set of seed URLs used to create the testbed were used as starting points for each spider, which ran until 100,000 Web pages had been visited. The performance of the spiders were measured by precision and recall, the most popular measures used in information retrieval research. The Web pages visited by each spider were also examined by medical experts in a user study. The experimental results showed that the proposed Hopfield Net Spider consistently achieved better performance than the BFS Spider and the PageRank Spider. The Hopfield Net Spider and BFS Spider also ran significantly faster than the PageRank Spider (Chau & Chen, 2002).

# 6 Future Research

The proposed approaches to text classification will be evaluated. Three hundred documents will be randomly chosen from the testbed, and manually classified by a human domain expert as “acceptable” and “not acceptable” for a medical search engine. The 300 pages will be classified into 2 sets: training data and testing data. All other Web pages in the testbed will also be accessible for content, neighbor and link analysis by the classifier during the training and testing phases.

The proposed feature-based NN approach and feature-based SVM approach will be compared with 2 benchmark approaches: (1) a TF*IDF approach, and (2) a keyword-based SVM approach. The TF*IDF approach is adopted from Baujard et al. (1998). TF*IDF score is calculated for those terms found in the medical lexicon. The scores for all documents in the training set are calculated. A threshold that best divides the documents into the 2 classes is determined. This threshold is then used for testing. The SVM approach will be adopted from Joachims (1998). A subset of features (terms) are selected from the training documents using various techniques. Stop terms (e.g., a, the, is) will be filtered, and stemming will be applied using the Porter stemmer (Porter, 1980).

Each of the 4 approaches will be evaluated using 10-fold cross-validation. The results will then be compared. The results will be measured by precision, recall, and F-measure, which are commonly used in text classification evaluation. The effect of the number of training examples on the performance will also be analyzed.

# 7 References


Baujard, O., Baujard, V., Aurel, S., Boyer, C., and Appel, R. D. "Trends in Medical Information Retrieval on the Internet," *Computers in Biology and Medicine*, 28, 1998, pp. 589-601.

Brin, S. and Page, L. "The Anatomy of a Large-Scale Hypertextual Web Search Engine," in *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, Apr 1998.

Chau, M. and Chen, H. "Personalized and Focused Web Spiders," in *Web Intelligence*, N. Zhong, J. Liu Y. Yao (eds.), Springer, forthcoming.

Chakrabarti, S., Dom, B., and Indyk, P. "Enhanced Hypertext Categorization Using Hyperlink," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, Seattle, Washington, USA, Jun 1998.

Chakrabarti, S., Dom, B., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., and Kleinberg, J. "Mining the Web's Link Structure," *IEEE Computer*, (32:8), 1999, pp. 60-67.

Cho, J., Garcia-Molina, H., and Page, L. "Efficient Crawling through URL Ordering," in *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, Apr 1998.

Furnkranz, J. "Exploiting Structural Information for Text Categorization on the WWW," in *Proceedings of the 3rd Symposium on Intelligent Data Analysis (IDA'99)*, Amsterdam, Netherlands, pp. 487-497.

Hopfield, J. J. "Neural Network and Physical Systems with Collective Computational Abilities," *Proceedings of the National Academy of Science*, USA, (79:4), 1982, pp. 2554-2558.

Iwayama, M. and Tokunaga, T. "Cluster-based Text Categorization: A Comparison of Category Search Strategies," in *Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'95)*, pp. 273-281.

Joachims, T. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *Proceedings of the European Conference on Machine Learning*, Berlin, 1998, pp. 137-142.

Joachims, T., Chistianini, N., Shawe-Taylor, J. "Composite Kernels for Hypertext Categorization," in *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, 2001.

Kleinberg, J. "Authoritative Sources in a Hyperlinked Environment," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

Koller, D. and Sahami, M. "Hierarchically Classifying Documents Using Very Few Words," in *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, 1997, pp. 170-178.

Lam, S.L.Y, and Lee, D.L. "Feature Reduction for Neural Network Based Text Categorization," in *Proceedings of the International Conference on Database Systems for Advanced Applications (DASFAA '99)*, Hsinchu, Taiwan, Apr 1999.

Lewis, D. D. and Ringuette, M. "Comparison of Two Learning Algorithms for Text Categorization," in *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

Lyman, P. and Varian, H. R. "How Much Information," available at http://www.sims.berkeley.edu/how-much-info/, 2000/.

Manber, U., Smith, M., and Gopal, B. "WebGlimpse: Combining Browsing and Searching," in *Proceedings of the USENIX 1997 Annual Technical Conference*, Anaheim, California, Jan 1997.

Masand, B., Linoff, G., and Waltz, D. "Classifying News Stories Using Memory Based Reasoning," in *Proceeedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'92),* pp. 59-64.

McCallum, A., Nigam, K., Rennie, J., and Seymore, K. "A Machine Learning Approach to Building Domain-specific Search Engines," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999, pp. 662-667.

Ng, H. T., Goh, W. B., and Low, K. L. "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization," in *Proceedings of the 20th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'97)*, 1997, pp. 67-73.

Oh, H. J., Myaeng, S. H., and Lee, M. H. "A Practical Hypertext Categorization Method Using Links and Incrementally Available Class Information," in *Proceedigns of the 23rd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'00)*, 2000, pp. 264-271.

Porter, M.F. "An algorithm for suffix stripping," *Program*, 14(3), 1980, pp. 130-137.

Sumner, R. G., Jr., Yang, K., and Dempsey, B. J. "An Interactive WWW Search Engine for User-defined Collections," in *Proceedings of the 3rd ACM Conference on Digital Libraries*, Pittsburgh, Pennsylvania, USA, Jun 1998, pp. 307-308.

Vapnik, V. *Statistical Learning Theory*, Wiley, Chichester, GB, 1998.

Wiener, E., Pedersen, J. O., and Weigend, A. S. "A Neural Network Approach to Topic Spotting," in *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

Witten, I. H., Bainbridge, D., and Boddie, S. J. "Greenstone: Open-source DL Software," *Communications of the ACM*, 44(5), pp. 47.

Yang, Y. and Liu, X. "A Re-examination of Text Categorization Methods, in *Proceedings of the 22nd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 42-49.

Yang, Y., Slattery, S. and Ghani, R. "A Study of Approaches to Hypertext Categorization," *Journal of Intelligent Information Systems*, 18(2), March 2002.